

**ANALISIS SENTIMEN IMPOR BERAS 2018 PADA *TWITTER*
MENGUNAKAN METODE *SUPPORT VECTOR MACHINE*
DAN PEMBOBOTAN JUMLAH *RETWEET***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Renaza Afidianti Nandini

NIM: 145150207111104



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

ANALISIS SENTIMEN IMPOR BERAS 2018 PADA TWITTER MENGGUNAKAN
METODE *SUPPORT VECTOR MACHINE* DAN PEMBOBOTAN JUMLAH *RETWEET*

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Renaza Afidianti Nandini
NIM: 145150207111104

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2018

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I



Yuita Arum Sari, S.Kom., M.Kom

NIK. 201609 880715 2 001

Dosen Pembimbing II



Putra Pandu Adikara, S.Kom., M.Kom

NIP. 19850725 200812 1 002

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T., M.T., Ph.D.

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundangundangan yang berlaku (UU No.20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)

Malang, 28 Desember 2018



Renaza Afidianti Nandini

NIM: 145150207111104

ABSTRAK

Renaza Afidianti Nandini. 2018. Analisis Sentimen Impor Beras 2018 Pada Twitter Menggunakan Metode *Support Vector Machine* Dan Pembobotan Jumlah *Retweet*. Skripsi Program Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom dan Putra Pandu Adikara, S.Kom., M.Kom

Media sosial Twitter merupakan salah satu basis data real time terbesar dan sangat berguna untuk mengetahui persepsi masyarakat di Indonesia. Permasalahan polemik impor beras pada tweet Twitter menjadi hal yang penting untuk dikaji sebagai pemrosesan teks. Pada penelitian ini membahas analisis sentimen pada Twitter impor beras 2018 menggunakan metode *Support Vector Machine (SVM)* dan Pembobotan Jumlah *Retweet*. Penggunaan fitur pembobotan jumlah *retweet* menggunakan perbandingan nilai konstanta (α dan β) tertentu sebanyak 11 kali untuk memperoleh hasil analisis kelas positif dan negatif. Data yang digunakan pada penelitian ini sebanyak 318 data yang terdiri dari dua jenis data yaitu data latih dan data uji dengan perbandingan 70% data latih dan 30% data uji. Dari hasil pengujian akurasi dengan menggunakan metode *Support Vector Machine* tanpa pembobotan jumlah *retweet* sebesar 50,00%, *precision* sebesar 49,46%, *recall* sebesar 97,87%, dan *f-measure* sebesar 65,71%. Hasil pengujian akurasi dengan menggunakan metode *Support Vector Machine* dengan pembobotan jumlah *retweet* sebesar 50,00%, *precision* sebesar 49,46%, *recall* sebesar 01,00% dan *f-measure* sebesar 65,73%. Dapat disimpulkan bahwa penggunaan fitur pembobotan jumlah *retweet* dapat memberikan hasil yang cukup optimal dan mampu melakukan klasifikasi dalam analisis sentimen.

Kata kunci: *Twitter, analisis sentimen, Super Vector Machine, Pembobotan Jumlah retweet*

ABSTRACT

Renaza Afidianti Nandini. 2018. Analisis Sentimen Impor Beras 2018 Pada Twitter Menggunakan Metode *Support Vector Machine* Dan Pembobotan Jumlah *Retweet*. Skripsi Program Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom dan Putra Pandu Adikara, S.Kom., M.Kom

Social media Twitter is one of the largest real time databases and is very useful for knowing people's perceptions in Indonesia. The issue of rice import polemic on Twitter tweets is an important thing to study as text processing. This study discusses sentiment analysis on 2018 rice import Twitter using the Support Vector Machine (SVM) method and Weighting the Number of Retweets. The use of the weighting feature of the number of retweets uses a comparison of certain constants (α and β) 11 times to obtain the results of positive and negative class analysis. The data used in this study were 318 data consisting of two types of data namely training data and test data with a ratio of 70% training data and 30% test data. From the results of accuracy testing using the Support Vector Machine method without weighting the number of retweets by 50.00%, precision by 49.46%, recall by 97.87%, and f-measure by 65.71%. Accuracy testing results using the Support Vector Machine method with a weighting of retweet amount of 50.00%, precision of 49.46%, recall of 01.00% and f-measure of 65.73%. It can be concluded that the use of the weighting feature of the number of retweets can provide optimal results and is able to classify sentiment analysis.

Keywords: Twitter, sentiment analysis, Super Vector Machine, Weighting Number of Retweets

PRAKATA

Alhamdulillah rabbil'alamin. Puji dan syukur penulis ucapkan atas kehadiran Allah SWT atas ragmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan skripsi yang berjudul "Analisis Sentimen Impor Beras 2018 Pada Twitter Menggunakan Metode Support Vector Machine Dan Pembobotan Jumlah Retweet" dengan baik.

Dalam pengerjaan skripsi ini penulis mendapatkan banyak dukungan dan bantuan baik moral maupun material dari berbagai pihak. Oleh karena itu penulis mengucapkan banyak terima kasih kepada:

1. Ibu Yuita Arum Sari, S.Kom, M.Kom dan Bapak Putra Pandu Adikara, S.Kom, M.Kom selaku dosen pembimbing yang telah sabar dan memberi banyak pengarahan dan saran untuk skripsi penulis sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Agus Wahyu Widodo, S.T., M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak M. Tanzil Furqon, S.Kom., M.Comp.Sc selaku Sekretaris Jurusan Teknik Informatika.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Kedua orang tua dan kakak penulis, Ayahanda tercinta Indra Budi dan Ibunda tercinta Luluk Arnawati, Mas Reza serta keluarga yang telah memberikan dukungan baik moril ataupun material, motivasi, serta doa demi kelancaran pengerjaan skripsi ini.
7. Bapak Machrus Abadi, S.Pd, M.Pd selaku pakar yang telah bersedia memvalidasi data skripsi penulis.
8. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberi ilmunya kepada penulis selama perkuliahan di kampus ini.
9. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan menyelesaikan skripsi di Teknik Informatika Universitas Brawijaya.
10. Teman-teman seperjuangan yang telah banyak membantu penulis baik secara langsung maupun tidak langsung yaitu Ratna Tri Utami., Uswatun Hasanah., Rhevita W., Eka Novita S, Alpia Pentidari., Yunita Dwi A., Rusdy Hasan., Adhitya Wira., Ivan Yusri., Anam R., Hery., Febiko., dan yang lainnya.
11. Teman-teman Kos Watu Mujur 1/10 yang selalu mendukung dan mendoakan kelancaran skripsi penulis.
12. Seluruh pihak yang telah membantu kelancaran pengerjaan skripsi yang tidak dapat penulis sebutkan satu per satu.

Semoga Allah SWT senantiasa membalas segala kebaikan dan jasa yang telah diberikan kepada penulis. Pada intinya, skripsi ini membahas tentang sistem yang mampu menerapkan algoritme *Support Vector Machine* dan Pembobotan Jumlah *Retweet* untuk mengklasifikasikan analisis masyarakat terhadap impor beras pada Twitter berdasarkan analisis sentimen *tweet*.

Penulis menyadari bahwa skripsi ini memiliki banyak kekurangan baik pada format penulisan maupun isi bahasannya, oleh karena itu diharapkan saran dan kritik yang dapat membangun. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak baik untuk pembaca maupun penelitian selanjutnya.

Malang, 12 November 2018

Penulis

renazafidianandini@gmail.com



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
ABSTRACT	v
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR KODE PROGRAM	xv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Kepustakaan	6
2.2 Twitter	8
2.3 Text Mining	9
2.4 <i>Pre processing</i> Dokumen <i>Tweet</i>	9
2.4.1 Tokenisasi	10
2.4.2 <i>Cleaning</i>	10
2.4.3 <i>Case Folding</i>	10
2.4.4 <i>Filterisasi</i>	11
2.4.5 <i>Stemming</i>	11
2.5 Pembobotan Teks	12
2.5.1 <i>Term Frequency (tf)</i>	12
2.5.2 <i>Document Frequency (df)</i>	12
2.5.3 <i>Inverse Document Frequency (idf)</i>	12

2.5.4 Term Frequency-Inverse Document Frequency (tf-idf)	13
2.6 Analisis Sentimen	13
2.7 Support Vector Machine	13
2.7.1 Sequential Training SVM	15
2.8 Pembobotan Jumlah Retweet	17
2.9 Penggabungan Pembobotan Jumlah Retweet	17
2.10 Normalisasi Min-max	18
2.11 Evaluasi	18
BAB 3 METODOLOGI	20
3.1 Studi Kepustakaan	20
3.2 Tipe Penelitian	20
3.3 Partisipan Penelitian	20
3.4 Lokasi Penelitian	21
3.5 Pengumpulan Data	21
3.6 Perancangan	21
3.7 Implementasi	22
3.8 Pengujian	22
3.9 Kesimpulan dan Saran	22
BAB 4 Perancangan dan Implementasi	23
4.1 Deskripsi Umum Sistem	23
4.2 Pre Processing Teks	24
4.2.1 Tokenisasi	25
4.2.2 Cleaning	25
4.2.3 Case Folding	27
4.2.4 Filterisasi	27
4.2.5 Stemming	28
4.3 Pembobotan Kata	29
4.3.1 Menghitung Nilai TF	30
4.3.2 Menghitung Nilai WTF	31
4.3.3 Menghitung Nilai IDF	31
4.3.4 Menghitung TF-IDF	32
4.4 Klasifikasi teks Support Vector Machine	33

4.4.1 Menghitung Matriks Hessian	34
4.4.2 <i>Sequential Training</i>	35
4.4.3 Menghitung Bias	37
4.4.4 Menghitung $h(x)$	39
4.5 Pembobotan Jumlah <i>Retweet</i> (non-tekstual)	40
4.6 Penggabungan	41
4.7 Manualisasi <i>Support Vector Machine</i> dan Pembobotan Jumlah <i>Retweet</i>	42
4.7.1 Manualisasi Tokenisasi	43
4.7.2 Manualisasi <i>Cleaning</i>	44
4.7.3 Manualisasi <i>Case Folding</i>	46
4.7.4 Manualisasi <i>Filterisasi</i>	47
4.7.5 Manualisasi <i>Stemming</i>	48
4.7.6 Manualisasi Pembobotan Kata TF, DF, Wtf dan IDF	49
4.7.7 Manualisasi TF-IDF	53
4.7.8 Manualisasi <i>Support Vector Machine</i>	55
4.7.9 Manualisasi Penggabungan	61
4.8 Perancangan Pengujian	70
4.8.1 Perancangan pengujian pengaruh parameter pada algoritme <i>Support Vector Machine (SVM)</i> terhadap hasil akurasi	70
4.8.2 Perancangan pengujian pengaruh pembobotan jumlah <i>retweet</i> terhadap hasil akurasi	72
4.8.3 Perancangan pengujian pengaruh <i>Support Vector Machine</i> dan pembobotan jumlah <i>retweet</i> terhadap hasil akurasi	73
4.9 Implementasi Sistem	74
4.10 <i>Preprocessing</i> Teks	76
4.10.1 Implementasi Fungsi Tokenisasi	76
4.10.2 Implementasi Fungsi <i>Cleaning</i>	76
4.10.3 Implementasi Fungsi <i>Case Folding</i>	77
4.10.4 Implementasi Fungsi <i>Filterisasi</i>	78
4.10.5 Implementasi Fungsi <i>Stemming</i>	78
4.11 Pembobotan Teks	79
4.11.1 Implementasi Fungsi TF	79

4.11.2 Implementasi Fungsi WTF	79
4.11.3 Implementasi Fungsi IDF	80
4.11.4 Implementasi Fungsi TF-IDF	80
4.12 Klasifikasi SVM	81
4.12.1 Implementasi Matriks Hessian	81
4.12.2 Implementasi <i>Sequential Training</i>	81
4.12.3 Implementasi Hitung Bias	83
4.12.4 Implementasi Hitung $Sign(h(x))$	84
4.13 Implementasi Fungsi Normalisasi Jumlah <i>Retweet</i>	85
4.14 Implementasi Penggabungan Pembobotan Tekstual dan Non-Tekstual	85
4.15 Pengujian	86
4.15.1 Implementasi Fungsi Evaluasi	86
BAB 5 Hasil dan pembahasan.....	88
5.1 Skenario Pengujian	88
5.1.1 Pengujian Nilai Parameter γ (<i>learning rate</i>)	88
5.1.2 Pengujian Nilai Parameter λ (<i>lambda</i>)	89
5.1.3 Pengujian Nilai Parameter C (<i>complexity</i>)	91
5.1.4 Pengujian Nilai Parameter ϵ (<i>epsilon</i>)	92
5.1.5 Pengujian Pengaruh Pembobotan Jumlah <i>Retweet</i>	93
5.1.6 pengujian pengaruh <i>Support Vector Machine</i> dan pembobotan jumlah <i>retweet</i> terhadap hasil akurasi	95
BAB 6 Kesimpulan dan saran.....	98
6.1 Kesimpulan.....	98
6.2 Saran	99
DAFTAR REFERENSI	100
LAMPIRAN	103

DAFTAR TABEL

Tabel 2.1 Kajian Kepustakaan	7
Tabel 2.2 Ilustrasi Proses Tokenisasi	10
Tabel 2.3 Ilustrasi Proses <i>Cleaning</i>	10
Tabel 2.4 Ilustrasi Proses <i>Case Folding</i>	11
Tabel 2.5 Ilustrasi Proses <i>Filterisasi</i>	11
Tabel 2.6 Ilustrasi Proses <i>Stemming</i>	12
Tabel 2.7 Kernel <i>Support Vector Machine</i>	15
Tabel 2.8 Indikator <i>confusion matrix</i>	18
Tabel 4.1 Data Latih <i>tweet</i>	42
Tabel 4.2 Data Uji <i>tweet</i>	43
Tabel 4.3 Manualisasi Tokenisasi (Data Latih)	43
Tabel 4.4 Manualisasi Tokenisasi (Data Uji)	44
Tabel 4.5 Manualisasi <i>Cleaning</i> (Data Latih)	44
Tabel 4.6 Manualisasi <i>Cleaning</i> (Data Uji)	45
Tabel 4.7 Manualisasi <i>Case Folding</i> (Data Latih)	46
Tabel 4.8 Manualisasi <i>Case Folding</i> (Data Uji)	46
Tabel 4.9 Manualisasi <i>Filterisasi</i> (Data Latih)	47
Tabel 4.10 Manualisasi <i>Filterisasi</i> (Data Uji)	48
Tabel 4.11 Manualisasi <i>Stemming</i> (Data Latih)	48
Tabel 4.12 Manualisasi <i>Stemming</i> (Data Uji)	49
Tabel 4.13 Manualisasi TF dan DF	49
Tabel 4.14 Manualisasi Wtf dan IDF	51
Tabel 4.15 Manualisasi TF-IDF	53
Tabel 4.16 Perhitungan Kernel Seluruh dokumen (data latih)	56
Tabel 4.17 Perhitungan Matriks Hessian Seluruh dokumen (data latih)	57
Tabel 4.18 Hasil perhitungan E_i pada seluruh dokumen (data latih)	57
Tabel 4.19 Hasil perhitungan $\delta\alpha$ pada seluruh dokumen (data latih)	58
Tabel 4.20 Hasil perhitungan antara α pada seluruh dokumen (data latih)	58
Tabel 4.21 Hasil perhitungan E saat memperoleh iterasi maksimum pada seluruh dokumen (data latih)	58

Tabel 4.22 Hasil perhitungan $\delta\alpha$ saat mencapai iterasi maksimum pada seluruh dokumen (data latih).....	58
Tabel 4.23 Hasil perhitungan α saat mencapai iterasi maksimum pada seluruh dokumen (data latih).....	59
Tabel 4.24 Hasil <i>support vector</i> yang diperoleh.....	59
Tabel 4.25 Hasil perhitungan nilai b (bias).....	59
Tabel 4.26 Hasil perhitungan <i>testing Support Vector Machine</i> pada data uji	60
Tabel 4.27 Manualisasi Pembobotan Jumlah <i>Retweet</i>	60
Tabel 4.28 Manualisasi Penggabungan.....	61
Tabel 4.29 Perancangan pengujian parameter γ (<i>learning rate</i>)	71
Tabel 4.30 Perancangan pengujian parameter λ (<i>lambda</i>)	71
Tabel 4.31 Perancangan pengujian parameter C (<i>complexity</i>).....	71
Tabel 4.32 Perancangan pengujian parameter ε (<i>epsilon</i>)	72
Tabel 4.33 Perancangan pengujian pengaruh pembobotan jumlah <i>retweet</i> terhadap hasil akurasi.....	72
Tabel 4.34 Perancangan pengujian pengaruh Support Vector Machine dan pembobotan jumlah <i>retweet</i> terhadap hasil akurasi	73
Tabel 4.35 Daftar fungsi implementasi sistem	74
Tabel 5.1 Hasil pengujian nilai parameter γ (<i>learning rate</i>)	88
Tabel 5.2 Hasil pengujian nilai parameter parameter λ (<i>lambda</i>).....	90
Tabel 5.4 Hasil Pengujian Nilai Parameter C (<i>complexity</i>)	91
Tabel 5.5 Hasil Pengujian Nilai Parameter ε (<i>epsilon</i>)	92

DAFTAR GAMBAR

Gambar 1.1 Data pengguna twitter	2
Gambar 4.1 Gambaran Umum Sistem	23
Gambar 4.2 Alur Proses <i>Pre Processing</i> Teks	24
Gambar 4.3 Alur Proses Tokenisasi	25
Gambar 4.4 Alur Proses <i>Cleaning</i>	26
Gambar 4.5 Alur Proses <i>Case Folding</i>	27
Gambar 4.6 Alur Proses <i>Filterisasi</i>	28
Gambar 4.7 Alur Proses <i>Stemming</i>	29
Gambar 4.8 Alur Proses Pembobotan Kata	30
Gambar 4.9 Alur Proses Perhitungan Nilai TF	31
Gambar 4.10 Proses Perhitungan Nilai WTF	31
Gambar 4.11 Alur Proses Perhitungan Nilai IDF	32
Gambar 4.12 Alur Proses Perhitungan Nilai TF-IDF	33
Gambar 4.13 Alur Proses Klasifikasi <i>Support Vector Machine</i>	34
Gambar 4.14 Alur Proses Hitung Matriks Hessian	35
Gambar 4.15 Alur Proses <i>Sequential Training</i>	37
Gambar 4.16 Alur Proses Perhitungan Bias	38
Gambar 4.17 Alur Proses Hitung $h(x)$	39
Gambar 4.18 Pembobotan jumlah <i>Retweet</i> (non-tekstual)	40
Gambar 4.19 Alur Proses Penggabungan	41
Gambar 5.1 Pengaruh nilai parameter γ (<i>learning rate</i>)	89
Gambar 5.2 Pengaruh nilai parameter λ	91
Gambar 5.3 Pengaruh nilai parameter C (<i>complexity</i>)	92
Gambar 5.4 Pengaruh nilai parameter ϵ (epsilon)	93
Gambar 5.5 Pengujian Pengaruh Pembobotan Jumlah Retweet pada perbandingan nilai konstanta sebanyak 11 kali	95
Gambar 5.6 Confusion Matrix terhadap hasil akurasi	97
Gambar 5.7 Hasil akurasi dengan menggunakan fitur dan tanpa menggunakan fitur	97

DAFTAR KODE PROGRAM

Kode Program 4.1 Implementasi Tokenisasi.....	76
Kode Program 4.2 Implementasi <i>Cleaning</i>	76
Kode Program 4.3 Implementasi <i>Case Folding</i>	77
Kode Program 4.4 Implementasi <i>Stoplist</i> dan <i>Filterisasi</i>	78
Kode Program 4.5 Implementasi <i>Stemming</i>	78
Kode Program 4.6 Implementasi TF.....	79
Kode Program 4.7 Implementasi WTF	79
Kode Program 4.8 Implementasi IDF	80
Kode Program 4.9 Implementasi TF-IDF.....	80
Kode Program 4.10 Implementasi Matriks Hessian.....	81
Kode Program 4.11 Implementasi <i>Sequential Training</i>	82
Kode Program 4.12 Implementasi Hitung Bias	83
Kode Program 4.13 Implementasi Hitung Kemiripan	84
Kode Program 4.14 Implementasi Normalisasi Jumlah <i>Retweet</i>	85
Kode Program 4.15 Implementasi Penggabungan Pembobotan Tekstual dan Non-Tekstual	86
Kode Program 4.16 Implementasi Evaluasi	87

DAFTAR LAMPIRAN

Lampiran 1 Tabel Data Latih	103
Lampiran 2 Tabel Data Uji	104
Lampiran 3 Surat Kesiapan Pakar.....	105



BAB 1 PENDAHULUAN

1.1 Latar belakang

Beras merupakan sumber utama pemenuhan karbohidrat bagi mayoritas masyarakat di Indonesia. Konsumsi akan beras ini mengalami peningkatan secara signifikan dengan angka konsumsi per kapita mencapai 124,89 kg per tahun atau sudah diatas 80 persen (Ruslan, 2018). Dalam hasil laporan produksi Departemen Pertanian Amerika Serikat (USDA) tahun periode 2015/2016 melaporkan Indonesia menjadi Negara di urutan ketiga yang memproduksi beras terbanyak di dunia setelah Tiongkok dan India yaitu dengan perkiraan produksi 36,30 juta ton. Namun, hal tersebut tidak membuat Indonesia untuk berhenti melakukan kegiatan impor beras dan justru menjadi salah satu Negara importir terbesar dunia (Listiyarini, 2016). Adanya kebutuhan beras yang terus mengalami kelonjakan setiap tahunnya. Bila tidak mengimpor, maka berdampak pada kenaikan harga beras dalam negeri yang tinggi disebabkan jumlah permintaan lebih tinggi dibandingkan jumlah persediaan beras dalam negeri. Di sisi lain, impor dapat menjatuhkan harga beras di kalangan petani disebabkan harga mereka berpotensi dipermainkan oleh para tengkulak (Nursiyono, 2014). Beras termasuk komoditas utama untuk berkontribusi besar terhadap inflasi. Tercatat komoditas beras memberikan andil untuk inflasi nasional pada tahun 2017, yaitu sebesar 0,16. Sementara terhadap garis kemiskinan pada 2017, komoditas beras ini memiliki andil sebesar 18,80 persen di perkotaan dan di tingkat pedesaan sebesar 24,52 persen. Badan Pusat Statistik (BPS) melaporkan data inflasi terakhir di bulan Januari 2018 mencapai sebesar 0,62 persen. Adapun beras merupakan penyumbang inflasi terbesar dengan kontribusi yang mencapai 0,24 persen (Andreas, 2018). Dengan adanya hal tersebut banyak masyarakat yang mencurahkan pendapatnya di media sosial Twitter.

Media sosial Twitter merupakan salah satu basis data *real time* terbesar dan sangat berguna untuk mengetahui persepsi masyarakat di Indonesia. Hingga kuartal pertama di tahun 2017 pengguna Twitter di dunia mencapai 328 juta pengguna. Indonesia sendiri hingga Mei 2016, jumlah penggunaanya mencapai 24,2 juta (Huda, 2017). Pesan dalam Twitter dikenal dengan sebutan *tweet*. Oleh karena itu, *tweet* dapat dijadikan sebagai media bagi masyarakat Indonesia untuk menyampaikan pendapat dan opini atau bahkan hanya untuk berbagi informasi yang diperoleh kepada orang lain. Banyak kalangan orang dewasa maupun para remaja dengan berbagai macam profesi yang menggunakan media sosial ini. Twitter juga merupakan wadah dalam menyampaikan berbagai keluhan, kesah, argumen, kritik dan saran terhadap berbagai aspek baik dalam lingkup sosial, politik, jasa, pangan dan teknologi. Salah satu aspek yang sering diperbincangkan adalah berbagai masalah pangan yang terjadi di Indonesia.



Gambar 1.1 Data pengguna twitter

Sumber: Setara.net (2018)

Impor Beras merupakan masalah pangan yang masih menjadi pro kontra hingga saat ini. Dengan berbagai keluhan yang dirasakan baik pada petani, pedagang pasar maupun konsumsi masyarakat akan beras di Indonesia secara keseluruhan (Listiyarini, 2016). Beras yang diimpor Perum Bulog dari Vietnam, Thailand, dan India jumlahnya berkisar 56,2% dari izin impor yang diberikan oleh Kementerian Perdagangan, yaitu sebanyak 500 ribu ton (Reily, 2018). Keputusan pemerintah melakukan impor beras mengingat faktor ketersediaan stok dan rendahnya kualitas gabah menjadi tantangan yang harus segera diselesaikan dalam waktu dekat. Agar harga beras tidak terlalu melambung di seluruh daerah. Di samping itu cadangan beras yang terdapat di Badan Urusan Logistik (Bulog) semakin menipis disertai dengan adanya efek fenomena el nino yang berpengaruh terhadap produksi pertanian, Sehingga kenaikan harga beras sangat berdampak kepada laju inflasi. Permasalahan polemik impor beras pada tweet Twitter menjadi hal yang penting untuk dikaji sebagai pemrosesan teks. Dengan berbagai macam reaksi masyarakat. Khususnya petani yang akan merugi karena harga gabah akan turun (Handayani, 2018). Maka dari itu, sangat penting bagi pemerintah yang mempunyai kewenangan penuh. Untuk memahami analisis masyarakat sebagai

gambaran keefektifan keputusan pemerintah dalam merealisasikan kegiatan impor beras dan dampak yang ditimbulkan pada polemik yang terjadi.

Agar dapat mengetahui analisis sentimen impor beras pada *tweet* Twitter, maka dibutuhkan analisis sentimen pada *tweet-tweet* yang berhubungan dengan tanggapan masyarakat yang berupa keluhan, komentar, informasi atau saran terhadap impor beras tersebut. *Opinion Mining* atau Analisis Sentimen adalah studi komputasional melalui opini-opini masyarakat, sentimen, dan emosi melalui entitas dan atribut yang dimiliki yang diekspresikan dalam bentuk teks (Liu, 2012).

Penelitian yang terkait pada media sosial Twitter yaitu dilakukan oleh Ahuja et al., (2017) peneliti melakukan klasifikasi status emosional untuk model berbasis Twitter dengan mengklasifikasikan *tweet* Twitter ke dalam 8 kategori emosi yang berbeda dengan membandingkan beberapa metode klasifikasi yaitu *Naïve Bayes*, *K-Nearest Neighbours* (KNN), *Support Vector Machine* (SVM) dan *Decision Tree* (ID3). Didapatkan hasil akurasi terbaik memperoleh nilai 90% dengan menggunakan SVM-linier. Penelitian lain yang membahas *opinion mining* seperti penelitian yang dilakukan oleh Kristiyanti (2015) dengan melakukan analisis sentimen review produk kosmetik pada situs web amazon.com dengan menggunakan algoritme *Support Vector Machine* dan *Particle Swarm Optimization* untuk seleksi fitur. Didapatkan algoritme SVM berbasis PSO memperoleh nilai akurasi lebih tinggi dibandingkan hanya menggunakan algoritme SVM yaitu memperoleh nilai akurasi sebesar 97% dengan selisih akurasi 8%.

Penelitian lain yang serupa yaitu penelitian yang dilakukan oleh Luqyana (2018) yang menggunakan metode *Support Vector Machine* dengan menggunakan fitur *Lexicon Based* untuk analisis sentimen *cyberbullying* pada Instagram. Didapatkan hasil pengujian akurasi terbaik mencapai 90% dengan menggunakan *kernel SVM polynomial* berderajat 2. Fitur yang akan digunakan pada penelitian ini yaitu menggunakan fitur pembobotan jumlah *retweet*. Fitur ini mampu memberikan pengaruh hasil nilai klasifikasi dan persentase akurasi sistem jika nilai konstanta α dan β tepat (Estu, 2017). Berdasarkan penelitian tersebut maka penelitian ini akan membahas mengenai analisis sentimen impor beras 2018 pada Twitter menggunakan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet*. Dengan penelitian ini diharapkan pemerintah yang mempunyai kewenangan penuh dapat mengetahui kondisi yang terjadi di masyarakat dengan adanya keputusan impor beras sehingga mampu menyelesaikan secara konkret jika ada pihak yang merasa dirugikan.

1.2 Rumusan masalah

Pada uraian latar belakang tersebut dapat dipaparkan rumusan masalah pada penelitian ini, antara lain:

1. Bagaimana implementasi algoritme *Support Vector Machine* dan Pembobotan Jumlah *Retweet* dalam membantu analisis sentimen pada *tweet* Twitter?
2. Bagaimana pengaruh Pembobotan Jumlah *Retweet* dalam melakukan klasifikasi kelas sentimen pada Twitter berdasarkan analisis sentimen *tweet*?

3. Bagaimana hasil akurasi klasifikasi sentimen positif dan negatif berdasarkan analisis sentimen impor beras 2018 pada Twitter menggunakan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet*.

1.3 Tujuan

Pada uraian latar belakang dan perumusan masalah yang sudah dipaparkan, tujuan pada penelitian ini antara lain:

1. Menerapkan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet* untuk menganalisis kelas sentimen positif dan negatif terhadap analisis sentimen pada *tweet* Twitter?
2. Mengetahui pengaruh Pembobotan Jumlah *Retweet* dalam melakukan klasifikasi kelas sentimen pada Twitter berdasarkan analisis sentimen *tweet*?
3. Menguji tingkat akurasi klasifikasi sentimen positif dan negatif berdasarkan analisis sentimen impor beras 2018 pada Twitter menggunakan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet*.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat antara lain:

1. Menambah pengetahuan untuk mengetahui hasil kelas sentimen positif dan negatif berdasarkan analisis sentimen *tweet*.
2. Menambah pengetahuan pengaruh Pembobotan Jumlah *Retweet* terhadap hasil klasifikasi kelas sentimen pada Twitter analisis sentimen *tweet*.
3. Mengetahui tingkat akurasi hasil klasifikasi sentimen positif dan negatif berdasarkan analisis sentimen impor beras 2018 pada Twitter menggunakan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet*.

1.5 Batasan masalah

Batasan masalah yang digunakan untuk mengarahkan dan memberi ruang lingkup penelitian yang baik sehingga output dari penelitian memberikan hasil yang optimal dan efektif. Batasan yang diterapkan adalah sebagai berikut:

1. *Algoritme* yang digunakan untuk pengklasifikasian adalah *Support Vector Machine* (SVM).
2. Fitur yang diterapkan adalah fitur *Term Frequency-Inverse Document Frequency* (TF-IDF) dan fitur pembobotan jumlah *retweet* berdasarkan jumlah *retweet* pada data latih dan data uji
3. Obyek dalam penelitian ini adalah dokumen teks berupa *tweet* bahasa Indonesia di media sosial Twitter. *Tweet* untuk data latih dan data uji.
4. Pesan *tweet* yang digunakan berasal dari pengguna Twitter di Indonesia yang membuat status *tweet* tentang impor beras 2018.
5. Data yang digunakan adalah historis pesan *tweet* selama 2 bulan dari tanggal 14 Maret 2018 sampai dengan 14 Mei 2018.

6. Data yang digunakan berjumlah 318 data yang terdiri dari data latih dan data uji. Data latih dan data uji diambil melalui <https://twitter.com/search-home> yang diambil sendiri oleh penulis.

1.6 Sistematika pembahasan

Untuk menjelaskan gambaran mengenai skripsi ini, berikut diuraikan pembahasan isi pada keseluruhan penulisan skripsi untuk setiap bab, sebagai berikut:

BAB I	PENDAHULUAN Berisi uraian mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan manfaat, serta sistematika penyusunan berdasarkan penelitian yang diangkat.
BAB II	LANDASAN KEPUSTAKAAN Menguraikan topik penelitian berdasarkan teori yang diangkat untuk menjadi acuan dasar dalam pembuatan sistem pada penelitian ini.
BAB III	METODOLOGI PENELITIAN Menguraikan suatu gambaran umum tentang tahap yang dilakukan serta metode diterapkan, meliputi studi literatur, pengumpulan data, analisis dan perancangan sistem, implementasi, pengujian, analisis dan kesimpulan.
BAB IV	PERANCANGAN DAN IMPLEMENTASI Berisi uraian perancangan sistem dan implementasi. Bab ini membahas tahap-tahap yang dilakukan untuk implementasi berdasarkan metodologi dan perancangan sistem yang dibahas sebelumnya.
BAB V	HASIL DAN PEMBAHASAN Berisi uraian tahap-tahap pada uji coba sistem dan memperoleh suatu analisa melalui pengujian yang di uji coba.
BAB VI	KESIMPULAN DAN SARAN Berisi uraian kesimpulan dan saran dari keseluruhan hasil pada penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas berdasarkan hasil penelitian yang didapatkan sebelumnya yaitu berkaitan pada permasalahan yang diterapkan. Serta membahas teori-teori pendukung untuk diterapkan pada penelitian ini.

2.1 Kajian Kepustakaan

Pada kajian kepastakaan ini membahas penelitian-penelitian yang telah dilakukan sebelumnya, untuk mendukung penelitian penulis. Penelitian terkait dengan objek Twitter. Seperti penelitian Ahuja, et al., (2017) untuk mendeteksi secara akurat keadaan emosional seseorang pada media sosial Twitter yaitu mengklasifikasikan *tweet* Twitter ke dalam delapan kategori emosi yang berbeda. Untuk data setiap tiap kelas emosi didapatkan melalui kamus yang diperoleh dari National Research Council, Kanada. Kemudian melakukan *preprocessing tweet* dengan cara menghilangkan kata maupun istilah yang terdapat *URL*, tanda ('), bahasa gaul dan kata yang bersifat repetitif seperti 'happpppyyy' setelah *tweet* diproses dan dibersihkan dengan tepat. Selanjutnya di *vector* oleh kata-kata identifikasi yang termasuk ke dalam salah satu delapan kategori emosi. Peneliti membuat *vector* dari data uji dan pelatihan dengan menetapkan perbandingan empat metode klasifikasi yaitu metode *Naïve Bayes*, *Support Vector Machine*, *K-Nearest Neighbours* dan *Decision Tree*. Dari penelitian tersebut memperoleh nilai akurasi tertinggi sebesar 90% dengan menggunakan metode SVM, 85,5% menggunakan metode KNN, 86% menggunakan metode *Decision Tree*, dan 82% menggunakan metode *Naïve Bayes*.

Pada penelitian yang terkait dengan *opinion mining* adalah penelitian yang dilakukan oleh Kristiyanti (2015) tentang analisis sentimen *review* produk dengan menggunakan *algoritme Support Vector Machine*. Dalam penelitian ini mengumpulkan 100 data positif *review* produk kosmetik dan 100 data *review* produk negatif produk kosmetik yang didapatkan dari situs *web* amazon.com. Penelitian ini memakai 20 kata yang berkaitan dengan sentimen produk kosmetik seperti *great*, *perfect*, *good*, *excellent*, *favorite*, *liked*, *nice*, *love*, *best*, *awesome*, *hate*, *horrible*, *fake*, *disgusting*, *disappointed*, *worse*, *ridiculously*, *crap*, *weird*, dan *awful*. Pengklasifikasian penelitian ini dikelompokkan menjadi dua kelas yaitu positif dan negatif. Serta menerapkan *algoritme Particle Swarm Optimization* (PSO) sebagai seleksi fitur untuk melakukan klasifikasi. Hasil akurasi yang diperoleh dari penelitian ini sebesar 89,00% dengan menggunakan *algoritme Support Vector Machine*. Sedangkan hasil pengujian yang didapatkan ketika menggunakan *Support Vector Machine* berbasis *Particle Swarm Optimization* (PSO) dapat meningkatkan nilai akurasi sebesar 97,00%.

Selanjutnya, Penelitian lain yang serupa terkait dengan *opinion mining* dan mengimplementasikan metode SVM yaitu penelitian yang dilakukan oleh Luqyana (2018) mengenai analisis sentimen tindakan *Cyberbullying* pada komentar media sosial *Instagram* untuk mengetahui berbagai komentar yang bersifat negatif dan komentar yang bersifat positif. Dengan dilakukan pengambilan data secara *offline*

sebanyak 400 komentar yang dibagi dengan perbandingan 70% data latih dan 30 data uji. Penelitian ini mengimplementasikan fitur *Lexicon Based* dalam tiap komentar dengan menggunakan kamus yang berisi sentimen kata positif dan kata negatif. Akurasi yang didapatkan memperoleh nilai sebesar 87% dengan menggunakan fitur *Lexicon Based* dan akurasi dapat meningkat sebesar 90% tanpa menggunakan fitur *Lexicon Based* dengan perbandingan komposisi data 50:50 dengan nilai parameter kernel *polynomial degree* = 2 didapatkan hasil akurasi paling terbaik mencapai 80%. Penelitian selanjutnya yang dilakukan Perdana dan Pinandito (2017) melakukan analisis sentimen terhadap pemilihan nominasi Gubernur Jakarta 2017. Data didapatkan melalui API Twitter sebanyak 14367 *tweet* pada bulan Oktober 2016 dengan perbandingan rasio 3:1 untuk data latih dan data uji. Penelitian ini menerapkan fitur kombinasi tekstual dan non-tekstual (*Likes-Retweet*) yang berguna untuk membantu melakukan prediksi sentimen. Penerapan *Algoritme Naïve Bayes* dilakukan untuk klasifikasi tekstual, Sedangkan untuk melakukan klasifikasi non-tekstual menggunakan *Fisher Score* pada penelitian ini. Hasil *F-Measure* yang diperoleh melalui kedua kombinasi fitur tersebut sebesar 0,838% dengan kombinasi terbaik dari prediksi klasifikasi tekstual dan non-tekstual (*Likes-Retweet*) yaitu 0,6 dan 0,4.

Penelitian lain yang menerapkan fitur non-tekstual yaitu Penelitian yang dilakukan oleh Estu (2017) tentang analisis sentimen terhadap tayangan televisi berdasarkan opini masyarakat. Data penelitian tersebut didapatkan melalui media sosial Twitter sejumlah 400 *tweet* dalam bahasa Indonesia. *Algoritme* yang diterapkan yaitu *K-Nearest Neighbours* dengan adanya penambahan fitur pembobotan jumlah *retweet* sebagai proses klasifikasi. Tahap pertama yang dilakukan adalah pembobotan secara tekstual, pembobotan jumlah *retweet* (non-tekstual) yang sudah melakukan normalisasi *min-max*, Kemudian melakukan penggabungan antara keduanya, Sehingga mampu mengetahui dokumen bernilai positif atau negatif. Hasil penelitian ini memperoleh nilai tingkat akurasi sebesar 82,50% dengan menggunakan pembobotan tekstual. Sedangkan ketika menggunakan fitur tambahan pembobotan jumlah *retweet* akurasi meningkat menjadi 83,33% dengan didapatkan hasil pengujian nilai konstanta optimal $\alpha=0,8$ dan $\beta=0,2$ serta nilai $k=3$. Kajian pustaka yang merujuk terhadap penelitian-penelitian terkait pada Tabel 2.1.

Tabel 2.1 Kajian Kepustakaan

No.	Penelitian	Objek dan Metode	Hasil Penelitian
1	(Ahuja, et al., 2017)	Mendeteksi keadaan emosional seseorang pada Twitter dengan membandingkan metode <i>Naïve Bayes</i> , <i>Support Vector Machine</i> , <i>K-Nearest</i>	Melalui pengujian empat metode diperoleh nilai akurasi 90% pada metode <i>Support Vector Machine</i> , 85.5% metode KNN, 86% metode <i>Decision Tree</i>

		<i>Neighbours</i> dan <i>Decision Tree</i> .	dan 82% metode <i>Naïve Bayes</i> .
2	(Kristiyanti, 2015)	Review Produk Kosmetik pada situs <i>web</i> amazon.com menggunakan metode Support Vector Machine dan Particle Swarm Optimization Sebagai Metode Seleksi Fitur.	Klasifikasi pada dua kelompok kelas positif dan kelas negatif. Hasil pengujian akurasi memperoleh 97% pada metode <i>Support Vector Machine</i> berbasis <i>Particle Swarm Optimization</i> (PSO).
3	(Luqyana, 2018)	Tindakan <i>Cyberbullying</i> pada komentar <i>Instagram</i> menggunakan metode <i>Support Vector Machine</i> .	Hasil pengujian analisis sentimen berupa komentar positif dan negatif memperoleh nilai akurasi hingga 90 % menggunakan kernel <i>polynomial degree 2</i> .
4	(Estu, 2017)	<i>Opini</i> masyarakat terhadap tayangan televisi pada Twitter dengan menggunakan metode <i>K-Nearest Neighbours</i> dan <i>pembobotan jumlah retweet</i> .	Hasil analisis sentimen pada dokumen bernilai positif atau negatif. Hasil pengujian akurasi memperoleh 83,33% dengan penambahan fitur <i>pembobotan jumlah retweet</i> .
5	(Perdana & Pinandito, 2017)	Memprediksi pemilihan nominasi Gubernur Jakarta 2017 pada Twitter dengan menggunakan metode <i>Naïve Bayes</i> fitur kombinasi tekstual dan non-tekstual (<i>Likes-Retweet</i>)	Hasil penelitian memperoleh nilai <i>F-measure</i> yang cukup baik yaitu sebesar 0,838% menggunakan data sebanyak 14367 tweet Twitter.

2.2 Twitter

Fenomena *microblogging* di jaringan sosial *online* telah menjalar selama beberapa tahun terakhir. Twitter adalah situs *microblogging* yang secara umum para penggunanya mampu mengirimkan pesan teks atau *tweet* dengan batasan maksimum 280 karakter. (Perez, 2017) Pesan tersebut dapat dilihat oleh seluruh

pengguna internet, namun dapat juga dibatasi oleh pengirimnya sehingga hanya pengguna lain yang menjadi pengikut dari akun pengirim pesan.

Twitter memiliki pengguna aktif bulanan sebanyak 328 juta. Menghasilkan petabyte data secara teratur pada pengguna Twitter di seluruh dunia. Melalui hasil dari popularitas global, situs jejaring sosial ini menjadi bagian penting dalam kehidupan sehari-hari. Data Statistik terbaru menunjukkan bahwa setiap hari memperoleh lebih dari 500 juta tweet, dengan volume yang semakin tinggi dalam beberapa kasus peristiwa yang sedang berlangsung di dunia nyata (Amjad, et al., 2017) Dengan munculnya Internet broadband seluler, telah memberikan kenyamanan untuk mengekspresikan pandangan seseorang mengenai masalah sosial, politik, dan ekonomi di media sosial kapan pun dan di mana saja.

Namun tidak di pungkiri juga banyak masyarakat menggunakannya terutama untuk berbagi atau menemukan tautan hal-hal yang menarik minat mereka. Twitter telah menjadi pemain kunci dalam mengetahui segala topik yang sedang diperbincangkan hingga saat ini serta membantu menyebarkan media dan gagasan yang mereka minati.

2.3 Text Mining

Text mining adalah jalur akses informasi untuk dapat menghubungkan para pengguna dengan kurun waktu secara tepat dalam pemrosesan informasi yang lebih sedikit atau mampu melakukan transformasi informasi teks. Di sisi lain dapat memberikan akses untuk membantu menganalisis, memahami informasi dan memfasilitasi pengambilan keputusan pada suatu data teks maupun *query* penting yang tidak relevan. Secara teknis, teknik ini berfokus pada model utama, algoritme, dan aplikasi tentang apa yang dapat dipelajari dari berbagai jenis data teks (Aggarwal & ChengXiang, 2012). Perbedaan mendasar dengan Data Mining pada umumnya, Text Mining mengolah data teks yang tidak terstruktur, maka proses text mining memerlukan beberapa tahap awal (*preprocessing*) yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.

Manfaat dari *Text Mining* menghasilkan inovasi untuk membantu masyarakat lebih memahami dan memanfaatkan informasi dalam penyimpanan dokumen. Pengembangan teknologi baru guna untuk mengatasi masalah seperti pendeteksian topik, pelacakan, dan trending. Mengidentifikasi topik yang muncul dalam korpus teks melalui mesin secara otomatis, agar memiliki dampak yang signifikan dalam sistem informasi seperti mesin pencari dan sistem manajemen dokumen (Srivastava & Sahami, 2009).

2.4 Pre processing Dokumen Tweet

Pre Processing dokumen *tweet* adalah tahapan awal dari text mining. Pada tahapan ini dokumen *tweet* yang telah diseleksi akan diolah, agar membuat dokumen yang dimasukkan lebih konsisten untuk memfasilitasi representasi teks. Dari data pesan *tweet* yang tidak terstruktur akan diolah menjadi data terstruktur yang siap diolah pada tahap berikutnya. Tahapan *Pre Processing* dokumen *tweet*

yang digunakan pada penelitian ini antara lain tokenisasi, *data cleaning*, *case folding*, *filterisasi* dan *stemming*.

2.4.1 Tokenisasi

Tokenisasi digunakan untuk melakukan pemecahan pesan *tweet* yang terkumpul menjadi kumpulan *token* atau kumpulan kata-kata. Dalam proses ini dokumen *tweet* akan dipisah menjadi kumpulan kata. Contoh tahap proses tokenisasi dapat dilihat pada Tabel 2.1. Pesan *tweet* yang ada memiliki 25 spasi dan akan dipecah menjadi 25 *token*. Proses ini memudahkan proses selanjutnya.

Tabel 2.2 Ilustrasi Proses Tokenisasi

Data Tweet	Proses Tokenisasi
Negeri yang maju harus rajin impor, biar kapitalis semakin makmur dan kemiskinan semakin menjamur. Sehingga lambat laun bak sapi mati kelaparan diatas rerumputan. #Importerus #imporberas	['Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor,', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', '#Importerus', '#imporberas']

2.4.2 Cleaning

Cleaning digunakan untuk pembersihan data pesan *tweet* pada bagian yang tidak diperlukan pada proses berikutnya atau tidak memiliki pengaruh dalam melakukan pemrosesan teks seperti *username* (@), (http), (#), dan *Retweet* (RT). Ilustrasi pada tahap proses *Cleaning* dapat dilihat pada Tabel 2.2.

Tabel 2.3 Ilustrasi Proses Cleaning

Data Tweet	Proses Cleaning
'Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor,', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', '#Importerus', '#imporberas'	'Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor,', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', 'Importerus', 'imporberas']

2.4.3 Case Folding

Case folding digunakan untuk melakukan penyamaan bentuk huruf pada setiap kata. Dengan adanya proses ini semua huruf yang ada dapat diubah bentuknya menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dianggap pemisah kata (*delimiter*). Contoh tahap proses *case folding* dapat ditunjukkan pada Tabel 2.3 dapat dilihat bahwa dalam kata "Negeri"

berubah menjadi kata “negeri”. Proses ini tidak merubah arti dari kata namun harus dilakukan untuk mempersiapkan data pada proses berikutnya yaitu pada proses *filterisasi*. Dengan proses ini diharapkan setiap kata yang sama dihitung sebagai satu *token*. Tanpa proses ini setiap kata yang sama bisa dianggap sebagai *token* yang berbeda karena susunan bentuk hurufnya yang berbeda.

Tabel 2.4 Ilustrasi Proses Case Folding

Data Tweet	Proses Case Folding
'Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', 'Importerus', 'imporberas'	'negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas'

2.4.4 Filterisasi

Filterisasi atau *stopword removal* merupakan penghapusan *token* yang tidak mempunyai makna tidak penting karena hal tersebut tidak memiliki pengaruh pada arti dalam *tweet*. Contohnya seperti kata “biar”. Langkah ini bertugas untuk menyamakan kata dari token ke kata-kata *stopword* pada file *stoplist.txt* kemudian token yang cocok akan dihilangkan. Daftar *stoplist* didapat dari data penelitian tala (2003). Pada Tabel 2.4 dapat dilihat bahwa dalam proses *filterisasi*, *token* “yang” dibuang karena tidak memiliki pengaruh pada arti pesan *tweet*.

Tabel 2.5 Ilustrasi Proses Filterisasi

Data Tweet	Proses Filterisasi
'negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas'	'negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'kemiskinan', 'menjamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas'

2.4.5 Stemming

Stemming melakukan tugas untuk mengubah kata imbuhan menjadi bentuk kata dasar dengan menghilangkan awalan dan akhiran dari sebuah kata. Pada penelitian ini akan digunakan library *stemming* Jsastrawi. Jsastrawi merupakan library stemming yang dikembangkan oleh Andy Librarian. Sastrawi menerapkan algoritme yang berbasis Nazief and Andriani, kemudian ditingkatkan oleh algoritma *Confix Stripping*, kemudian ditingkatkan lagi oleh algoritme ECS, lalu ditingkatkan lagi oleh Modified ECS (Librian, 2014). Untuk contoh tahap proses

stemming dapat dilihat pada Tabel 2.5 bahwa pada proses *stemming*, *token* "semakin" diubah menjadi "makin". Proses ini dilakukan untuk mengurangi jumlah *token*.

Tabel 2.6 Ilustrasi Proses Stemming

Data Tweet	Proses Stemming
'negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'kemiskinan', 'menjamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'impor beras'	'negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'miskin', 'jamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'lapar', 'atas', 'rumput', 'importerus', 'imporberas'

2.5 Pembobotan Teks

Pembobotan kata merupakan langkah yang dilakukan setelah tahap *Pre Processing*. Term yang terdapat pada hasil *Pre Processing* dijadikan indeks untuk mewakili dokumen agar memberikan kemudahan pada proses selanjutnya. Pembobotan teks merupakan tahap untuk memberi nilai berdasarkan term indeks. Berikut beberapa langkah memberikan bobot pada dokumen teks, antara lain:

2.5.1 Term Frequency (tf)

Term frequency $tf_{t,d}$ merupakan term t dalam dokumen d didefinisikan adanya suatu term t di dokumen d . Seperti halnya binary, hanya saja mempertimbangkan jumlah kemunculan kata pada dokumen *count vector*. Nilai bobot pada *term* akan semakin tinggi jika *term* yang muncul semakin banyak dalam suatu dokumen. Pada proses selanjutnya, nilai TF akan dihitung dengan rumus perhitungan algoritma atau disebut *weighting term frequency* (W_{tf}). Persamaan 2.1 menunjukkan rumus perhitungan.

$$tf_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

2.5.2 Document Frequency (df)

Document frequency (df_t) adalah jumlah dokumen yang terdapat term t . Pada umumnya terdapat di banyak dokumen atau suatu nilai *term frequency*-nya.

2.5.3 Inverse Document Frequency (idf)

Apabila pembobotan Kata-kata yang muncul di banyak dokumen adalah kata yang "tidak penting", *inverse document frequency* merupakan *term* yang paling tidak banyak muncul di dokumen adalah *term* yang mengandung bobot paling tinggi.

$$idf_t = \log_{10}(N/df_t) \quad (2.2)$$

2.5.4 Term Frequency-Inverse Document Frequency (tf-idf)

Pembobotan ini melakukan perkalian dari bobot term *frequency* dan *inverse document frequency* dalam suatu *term*.

$$W_{t,d} = \log(1 + tf_{t,d}) \times \log_{10}(N/df_t) \quad (2.3)$$

atau

$$W_{t,d} = W_{tf_{t,d}} \times idf_t \quad (2.4)$$

2.6 Analisis Sentimen

Pertumbuhan dalam bidang *opinion mining* atau analisis sentimen terjadi pada para pengguna melalui media sosial di web, seperti ulasan, forum diskusi, blog, dan mikroblog, karena memiliki volume besar dalam data opini yang dicatat dalam bentuk digital. *Opinion mining* merupakan sebuah studi komputasional tentang sentimen, pendapat, emosi dan sikap seseorang. Persoalan yang menarik ini semakin penting dalam bisnis dan masyarakat. Sehingga menawarkan banyak tantangan penelitian disamping itu menjanjikan wawasan yang berguna bagi siapa pun yang tertarik dalam melakukan *opinion mining* dan analisis media sosial (Liu, 2015). *Opinion mining* umumnya mengumpulkan polaritas dari teks yang ada dalam kalimat atau dokumen untuk mengetahui pendapat yang dikemukakan dalam kalimat atau dokumen tersebut apakah bersifat positif atau negatif. Klasifikasi data pada kelompok sentimen tertentu (positif atau negatif) dilakukan dengan membangun model probabilitas kemunculan suatu kata dalam dokumen yang telah dikelompokkan sebelumnya.

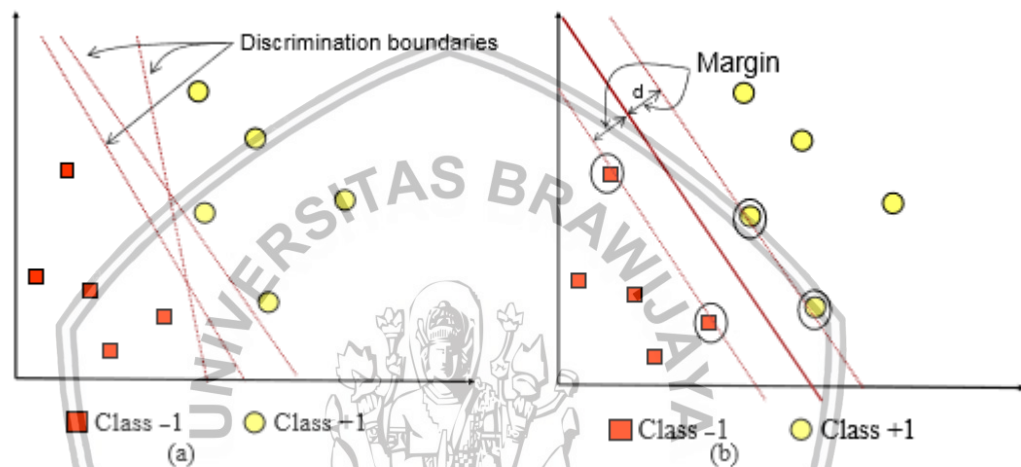
Popularitas teknik ini meningkat tajam sejak tahun 2001 melalui beberapa penelitian, Hingga saat ini telah terdapat ratusan penelitian di area ini yang sudah di publikasikan. Banyaknya pekerjaan untuk menganalisis sentimen dan pendapat dalam teks yang berorientasi politik biasanya berfokus pada sikap umum yang diekspresikan melalui teks yang belum pasti ditargetkan dalam masalah tertentu atau subjek yang tidak meluas (Pang & Lee, 2008). Selain pengelompokan data ke dalam kelompok sentimen, beberapa analisis juga menghitung skor numerik sentimen suatu data tekstual dengan menggunakan konsep pengurangan jumlah kemunculan kata negatif dengan kemunculan kata positif dalam suatu dokumen.

2.7 Support Vector Machine

Support Vector Machine (SVM) merupakan metode dengan mempertimbangkan tugas sederhana dalam klasifikasi. Umumnya digunakan untuk klasifikasi dan prediksi. Secara konseptual akan mencari *hyperplane* untuk memisahkan secara linier dengan melakukan pemetaan nonlinier yang tepat ke dimensi yang cukup tinggi, melalui data dari dua kelas mampu dipisahkan dengan *hyperplane* ini dengan menggunakan *support vector* dan *margin* kelas (Campbell & Yiming, 2011).

Model *Support Vector Machine* mewakili vektor atau titik di dalam ruang, dipetakan sehingga vektor dikelompokkan sebagai cluster dan celah pembagian di

antara dua cluster seluas mungkin. Kemudian klasifikasi dilakukan dengan cara memetakan vektor uji; vektor diprediksi berada pada kelas tertentu berdasarkan sisi mana celah itu. Meskipun waktu pelatihan SVM kebanyakan lambat, tetapi metode ini sangat akurat karena kemampuannya untuk menangani model-model nonlinier yang kompleks. SVM kurang rentan terhadap *overlifting* dibandingkan metode lainnya. SVM sangat cepat dan efektif untuk menyelesaikan masalah-masalah data teks. Data teks cocok untuk dilakukan klasifikasi dengan algoritma SVM karena sifat dasar teks yang cenderung mempunyai dimensi yang tinggi, dimana terdapat beberapa fitur yang tidak relevan, tetapi akan cenderung berkorelasi satu sama lain dan umumnya akan disusun dalam kategori yang terpisah secara linear (Aggarwal & Zhai, 2012).



Gambar 2.1 Ilustrasi SVM menemukan *hyperplane* terbaik yang mampu memisahkan kelas negatif dan positif

Sumber: (Nugroho, et al., 2003)

Pada Gambar 2.1 menjelaskan antara dua kelas data yang terpisah. Garis merah tebal merupakan *hyperplane* yang memisahkan dari dua titik, sehingga titik data pada satu sisi akan diberi label kelas negatif yaitu -1 dan label kelas positif yaitu +1. Titik data (*vector*) yang terdekat dengan *hyperplane* disebut dengan *support vector* pada garis merah kecil yang berpotongan dan memiliki pengaruh paling besar. Jarak antara *hyperplane* dengan *support vector* disebut dengan *margin*.

Pemodelan Support Vector Machine adalah sebagai berikut:

1. Titik pada data : $x_i = \{x_1, x_2, \dots, x_n\} \in R^n$
2. Kelas pada data : $y_i \in \{-1, +1\}$
3. Kumpulan data dan kelas : $\{(x_i, y_i)\}_{i=1}^N$
4. Fungsi keputusan klasifikasi $sign(h(x))$ dengan Persamaan 2.5.

$$f(x) = w \cdot x + b \quad \text{atau} \quad f(x) = \sum_{i=1}^m a_i y_i K(x, x_i) + b \quad (2.5)$$

Keterangan:

- N = banyaknya suatu data atau dokumen
- n = banyaknya suatu fitur
- w = parameter pada *hyperplane* yang dicari (garis *hyperplane* dan titik support vector antara garis yang tegak lurus)
- x = titik data input *Support Vector Machine*
- a_i = nilai bobot tiap data
- y_i = nilai bobot tiap kelas data
- $K(x, x_i)$ = fungsi kernel
- b = parameter *hyperplane* yang dicari (nilai bias)

Pada Tabel 2.7 menunjukkan fungsi kernel pada *Support Vector Machine*.

Tabel 2.7 Kernel Support Vector Machine

No	Kernel	Definisi Fungsi
1	Linier	$K(x, y) = x \cdot y$
2	<i>Polynomial of degree d</i>	$K(x, y) = (x \cdot y)^d$
3	<i>Polynomial of degree up to d</i>	$K(x, y) = (x \cdot y + c)^d$

Support Vector Machine merupakan klasifikasi untuk memberikan aturan yang dapat mengubah dokumen teks menjadi vektor sebelum diklasifikasi pada proses text mining.

2.7.1 Sequential Training SVM

Sequential Training SVM merupakan suatu metode yang bertugas untuk memperoleh *hyperplane*. Metode ini mampu memberikan solusi optimal dan mempercepat proses iterasi dibandingkan dengan menggunakan solusi konvensional (Vijayakumar & Wu, 1999). Adapun beberapa langkah untuk melakukan metode *sequential training*:

1. Melakukan inisialisasi terhadap parameter yang diaplikasikan, yaitu $\lambda, \gamma, C, \varepsilon$ dan iterasi maksimum. Misalkan $\lambda = 5$ (variabel scalar), $\gamma = 5$ (*learning rate*), $C = 1$ (variabel *slack*), $\varepsilon = 0.0001$ (*epsilon*) dan iterasi maksimum = 50.
2. Tahap berikutnya yaitu melakukan menginisialisasi $\alpha_i = 0$, lalu menghitung matriks hessian D_{ij} , dengan Persamaan 2.6.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.6)$$

Keterangan:

- D = Matriks Hessian
- y = kelas pada data

- x = data ke-
- K = Kernel
- λ = variable skalar

3. Selanjutnya melakukan tahap a dengan Persamaan 2.7, tahap b dengan Persamaan 2.8, dan tahap c dengan Persamaan 2.9, dengan $i=1$ hingga N . N adalah banyaknya jumlah data.

$$a) E_i = \sum_{j=1}^N \alpha_j D_{ij} \quad (2.7)$$

$$b) \delta \alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (2.8)$$

$$c) \alpha_i = \alpha_i + \delta \alpha_i \quad (2.9)$$

Keterangan:

- E = menghitung nilai Error
- $\delta \alpha$ = variabel tunggal
- γ = variabel *learning rate*
- C = variabel *slack*
- α = variabel *Lagrange Multiplier*

4. Melakukan tahap tiga hingga mencapai iterasi maksimum atau $\max(|\delta \alpha_i|) < \varepsilon$.
5. Pada tahapan yang sudah dilakukan sebelumnya dapat diperoleh α dan *support vector* (SV), yaitu data yang mempunyai nilai $\alpha > 0$. Selanjutnya, lakukan hitung *bias* b yang diperoleh pada Persamaan 2.10.

$$b = -\frac{1}{2} \left(\sum_{i=0}^N \alpha_i y_i (K(x_i, x^-)) + \sum_{i=0}^N \alpha_i y_i (K(x_i, x^+)) \right) \quad (2.10)$$

Keterangan:

- b = Nilai bias
- α = *Lagrange Multiplier*
- y = kelas data ke-
- K = Kernel
- x_i = Data *training* ke-
- x^+ = Kelas positif yang mempunyai nilai α tertinggi pada data *training*
- x^- = Kelas negatif yang mempunyai nilai α tertinggi pada data *training*

6. Melakukan klasifikasi dokumen teks ke dalam kelas sentimen yang diperoleh pada Persamaan 2.11. Nilai $\text{sign}(h(x))$ bernilai -1, dapat diketahui dokumen teks tersebut masuk ke dalam klasifikasi kelas sentimen negatif.

Namun jika memperoleh hasil nilai $sign(h(x)) + 1$, maka dokumen teks tersebut masuk ke dalam klasifikasi kelas sentimen positif.

$$h(x) = \sum_{i=0}^N \alpha_i y_i K(x, x_i) + b \quad (2.11)$$

Keterangan:

- b = Nilai bias
- α = *Lagrange Multiplier*
- y = kelas data ke-
- K = Kernel
- x_i = Data *training* ke-
- x = Data *testing*

2.8 Pembobotan Jumlah *Retweet*

Retweet adalah salah satu fitur yang terdapat pada media sosial Twitter. Fitur ini memberikan kemudahan membagikan atau menyebarkan *tweet* pada pengguna lainnya. Pembobotan jumlah *retweet* memberikan tujuan untuk menambah nilai positif pada *tweet* yang mempunyai banyak *retweet*, sehingga mampu memberikan pengaruh nilai sentimen pada *tweet*, serta mengetahui secara rinci *tweet* apa saja yang terdapat nilai sentimen positif. Jumlah angka *retweet* dapat memperjelas *tweet* memiliki sentimen positif (Perdana & Pinandito, 2017). Jumlah *retweet* adalah data non-tekstual yang akan diterapkan setelah klasifikasi SVM selesai dilakukan. Bobot yang akan diberikan, berdasarkan jumlah *retweet* pada dokumen *tweet* tersebut. Tahap selanjutnya jumlah *retweet* akan dilakukan normalisasi menggunakan min-max, sehingga bobot yang diberikan seimbang dengan bobot teks hasil klasifikasi SVM.

2.9 Penggabungan Pembobotan Jumlah *Retweet*

Pada proses ini merupakan langkah untuk melakukan penggabungan jumlah *retweet* dengan kombinasi antara untuk mengetahui sentimen positif atau negatif dalam pembobotan tekstual dan non-tekstual pada proses klasifikasi (Perdana & Pinandito, 2017). Didapatkan melalui hasil klasifikasi teks menggunakan SVM sebagai pembobotan tekstual dan pembobotan jumlah *retweet* sebagai pembobotan non-tekstual yang sebelumnya melakukan perhitungan normalisasi menggunakan Min-max pada nilai konstanta sebagai nilai pembanding dalam pembobotan tekstual dan non-tekstual. Rumus untuk penggabungan pembobotan tekstual dan non-tekstual dapat ditunjukkan pada Persamaan 2.12 (Perdana & Pinandito, 2017).

$$Penggabungan = (\alpha \times \text{Text}) + (\beta \times \text{Non - Teks}) \quad (2.12)$$

Keterangan:

- α = Konstanta pengali untuk bobot tekstual

- $Teks$ = Hasil nilai yang terdapat pada klasifikasi
- β = Konstanta pengali untuk bobot non – tekstual
- $Non - Teks$ = Nilai bobot pada jumlah *retweet*

2.10 Normalisasi Min-max

Normalisasi Min-max merupakan proses transformasi yang berfungsi pada atribut berupa angka akan diskala ke dalam suatu ukuran yang lebih kecil, seperti antara -1 sampai 1 atau 0 sampai 1 (Junaedi, et al., 2011). Nilai maksimum yang digunakan yaitu 0,9 dan nilai minimum yaitu 0,1 (Rofiqoh, 2017)

Persamaan normalisasi Min-max dapat dilihat pada Persamaan 2.13.

$$V'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \quad (2.13)$$

Keterangan:

- V'_i = nilai data pada hasil normalisasi *min-max*.
- v_i = nilai untuk melakukan normalisasi.
- \min_A = nilai minimum untuk tiap data.
- \max_A = nilai maksimum untuk tiap data.
- new_min_A = nilai minimum yang diketahui melalui proses normalisasi (0,1).
- new_max_A = nilai maksimum yang diketahui melalui proses normalisasi (0,9).

2.11 Evaluasi

Evaluasi pada tahap ini dilakukan untuk menguji hasil klasifikasi untuk mengetahui nilai kebenaran pada sistem. Secara umum, mampu memberikan teknik klasifikasi mengenai sentimen dengan menggunakan empat indikator yaitu *Accuracy*, *Precision*, *Recall*, dan *F-Measure* (Desai & Mehta, 2016). Tabel 2.8 menunjukkan indikator yang dihitung berdasarkan *confusion matrix*.

Tabel 2.8 Indikator *confusion matrix*

	<i>Actual Positive Cases</i>	<i>Actual Negatif Cases</i>
<i>Predicted Positives</i>	<i>True Positive Cases (TP)</i>	<i>False Negatif Cases (FN)</i>
<i>Predicted Negatifs</i>	<i>False Positive Cases (FP)</i>	<i>True Negatif Cases (TN)</i>

$$\text{Accuracy} = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (2.14)$$

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (2.15)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (2.16)$$

$$\text{F - measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.17)$$

Keterangan:

- *TP (True Positive)* = Nilai prediksi dan nilai sebenarnya merupakan jumlah nilai di kategori positif.
- *FP (False Positive)* = Nilai yang memiliki prediksi kategori positif namun pada nilai prediksi memiliki jumlah nilai kategori positif.
- *TN (True Negatif)* = Nilai prediksi dan nilai sebenarnya merupakan nilai pada jumlah kategori negatif.
- *FN (False Negatif)* = Nilai prediksi pada jumlah kategori negatif namun memiliki kategori positif pada nilai sebenarnya.

Accuracy didefinisikan apabila seluruh keadaan saat dilakukan prediksi benar ketika diprediksi. Maka jika nilai *accuracy* memperoleh 100% menunjukkan bahwa kondisi tersebut memprediksi secara benar seperti kondisi aslinya. *Precision* didefinisikan sebagai ketepatan sistem dalam klasifikasi data dengan menghitung perbandingan jumlah kondisi benar positif terhadap semua kondisi yang diprediksi positif. *Recall* merupakan relevannya suatu sistem dalam klasifikasi data dengan menghitung jumlah kondisi yang benar positif terhadap semua kondisi yang asli positif. *F-measure* mendefinisikan rata-rata nilai yang serupa antara nilai *precision* dan *recall*.

BAB 3 METODOLOGI

Pada bab ini berisi tahapan metodologi untuk diteraplan pada penelitian ini. Tahapan tersebut adalah Studi Kepustakaan, Tipe Penelitian, Partisipan Penelitian, Lokasi Penelitian, Pengumpulan Data, Perancangan, Implementasi, Pengujian, Kesimpulan dan Saran.

3.1 Studi Kepustakaan

Bagian ini merupakan sarana untuk memahami dan mempelajari konsep - konsep terkait permasalahan yang diketahui. Sehingga diperlukan untuk mencari referensi yang relevan seperti teori yang meliputi pada pengolahan data, metode dan pedoman dalam penyelesaian masalah. Dalam penelitian ini literature diperoleh melalui berbagai jurnal, e-book, artikel dan beberapa penelitian terkait permasalahan yang diangkat, Berikut ini merupakan teori yang akan dipelajari dalam studi kepustakaan:

1. *Text Mining*.
2. *Pre Processing Teks*.
3. Pembobotan Kata.
4. *Opinion Mining*.
5. Metode *Support Vector Machine*.
6. Metode *Polynomial Degree 2 Support Vector Machine*.
7. Pembobotan jumlah *retweet*.

Selanjutnya setelah dilakukan pencarian teori-teori yang dibutuhkan pada penelitian ini maka akan disertakan dalam dokumen penelitian sebagai sumber teori.

3.2 Tipe Penelitian

Tipe penelitian non-implementatif merupakan tipe penelitian yang digunakan pada penelitian ini dengan pendekatan analitik. Pendekatan analitik merupakan pendekatan yang mempunyai tujuan untuk mendefinisikan tingkat hubungan dari antar bagian pada topik penelitian di dalam kondisi tertentu ketika diteliti. Pendekatan ini akan menghasilkan sebuah hasil analisis.

3.3 Partisipan Penelitian

Partisipan penelitian untuk memvalidasi data yang sudah diklasifikasi sebelumnya, penelitian ini melibatkan dosen Bahasa Indonesia untuk melakukan validasi data yang digunakan pada penelitian ini yaitu berupa *tweet* Twitter.

3.4 Lokasi Penelitian

Lokasi penelitian yang digunakan adalah Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya Malang. Karena laboratorium tersebut merupakan penunjang sarana untuk mengetahui pengaruh penggunaan fitur pembobotan jumlah *retweet* pada *opinion mining* untuk mengetahui persepsi masyarakat pada kicauan *tweet-tweet* di Twitter. Maka laboratorium komputasi cerdas sudah tepat untuk dijadikan sebagai lokasi penelitian.

3.5 Pengumpulan Data

Tahap untuk melakukan pengumpulan data dari objek yang akan diproses. Bertujuan untuk mengumpulkan data data yang nantinya akan dikaji pada penelitian ini. Data diperoleh dari situs <https://twitter.com/search-home> dengan menggunakan metode manual untuk mengumpulkan *tweet-tweet* berbahasa Indonesia yang mengandung *term* "impor beras", *hashtag* #imporberas, #beras, #tolakimporberas karena memudahkan penulis mengumpulkan data sesuai dengan topik yang dibahas dan *mention* kepada *user* @jokowi selaku akun resmi Presiden Republik Indonesia yang umumnya merupakan tempat sasaran keluhan, opini dan pujian para masyarakat Indonesia. Serta @kementan selaku akun resmi Kementerian Pertanian Republik Indonesia karena selaku pihak yang berkepentingan dalam terjadinya keputusan melakukan kegiatan impor beras. Jumlah data yang digunakan pada penelitian ini sebanyak 318 *tweet*.

3.6 Perancangan

Perancangan ini bertujuan untuk melakukan analisis kebutuhan pada sistem yang sudah ditetapkan. Sistem ini dirancang dengan menggunakan bahasa pemrograman *python*. Perancangan sistem yang akan dibuat terdiri dari data *tweet* pada media sosial Twitter yang disimpan ke dalam dokumen *notepad.txt* sebagai data latih dan data uji. Data tersebut digunakan sebagai rujukan dalam proses analisis menggunakan metode *Support Vector Machine*. Selanjutnya sistem akan menerima *inputan* berupa *tweet* tentang impor beras 2018. *Tweet* tersebut akan melakukan proses pada tahapan yang terdapat dalam teori *Text Mining*, Tahap *Pre Processing* hingga proses pembobotan kata. Setelah selesai melakukan tahap pembobotan kata, kemudian masuk pada tahap proses klasifikasi teks dengan menggunakan metode *Support Vector Machine*.

Tahap berikutnya akan dilakukan perhitungan normalisasi *Min-max* yang berguna untuk melakukan perhitungan bobot pada banyaknya jumlah *retweet* (non-tekstual) yang terdapat di dokumen. Tahap terakhir melakukan perhitungan penggabungan pembobotan teks (tekstual) melalui perolehan hasil nilai klasifikasi *Support Vector Machine* dan pembobotan jumlah *retweet* (non-tekstual). Proses penggabungan akan memperlihatkan suatu nilai, sehingga dapat mengetahui dokumen yang telah melakukan proses klasifikasi bernilai positif atau negatif.

3.7 Implementasi

Bagian implementasi ini merupakan penerapan metode *Support Vector Machine* dan Pembobotan Jumlah *Retweet* untuk melakukan klasifikasi pada sentimen kelas negatif atau kelas positif pada analisis impor beras di *Twitter*. Beberapa spesifikasi kebutuhan yang diuraikan dalam pembuatan sistem ini sebagai berikut:

- a. Spesifikasi Kebutuhan Perangkat Keras (Hardware) yaitu:
 - Prosesor Intel(R) Core(TM) i5-6200U CPU @2.80GHz.
 - Memori 4.00 GB.
 - Hardisk dengan kapasitas 1 TB.
- b. Spesifikasi Kebutuhan Perangkat Lunak (Software) yaitu:
 - Sistem operasi yang digunakan yaitu menggunakan sistem Operasi Microsoft Windows 10.
 - Bahasa Pemrograman python dan IDE SpyDer.

3.8 Pengujian

Pengujian ini memberikan suatu tujuan untuk dapat mengetahui apakah sistem sudah berjalan dengan baik sesuai spesifikasi kebutuhan yang telah melandasinya dan mengetahui keefektifan melalui metode yang diterapkan untuk *opinion mining*. *Support Vector Machine* merupakan metode yang akan diterapkan pada penelitian ini. Pada pengujian dan evaluasi, metode tersebut selanjutnya diuji akurasi dalam mengetahui persepsi masyarakat terhadap rencana impor beras 2018 pada *Twitter*.

3.9 Kesimpulan dan Saran

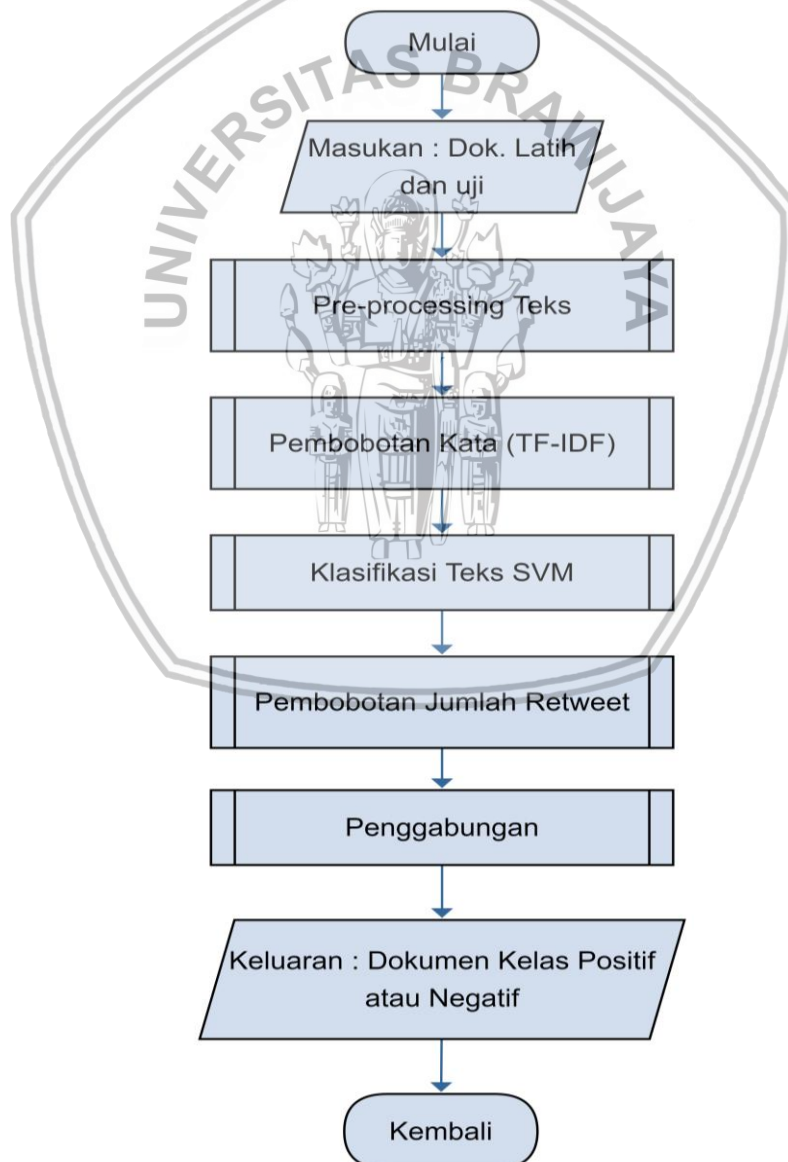
Kesimpulan akan dijelaskan setelah tahap perancangan, implementasi serta pengujian metode yang diimplementasikan pada penelitian ini selesai dilakukan, Serta kesimpulan didapatkan pada tahap pengujian sistem. Tahap akhir penulisan merupakan saran yang bertujuan untuk memberikan ulasan jika terdapat kesalahan, memaksimalkan kualitas isi penulisan serta memberikan pertimbangan untuk pengembangan penelitian selanjutnya.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bagian ini berisi tahapan rancangan seperti scenario flowchart maupun perhitungan manualiasi. Pada implementasi merupakan bagian untuk menjelaskan berjalannya alur sistem yang sudah dibuat.

4.1 Deskripsi Umum Sistem

Opinion Mining terhadap impor beras 2018 pada data *tweet* Twitter dengan menggunakan metode klasifikasi *Support Vector Machine* dan pembobotan jumlah *retweet* merupakan sistem yang dikembangkan untuk membantu menganalisis *tweet-tweet* pada Twitter yang mengandung persepsi negatif atau positif. Data yang digunakan pada penelitian ini merupakan data dari pencarian *tweet* Twitter yang telah diambil. Terdiri dari data latih dan data uji.



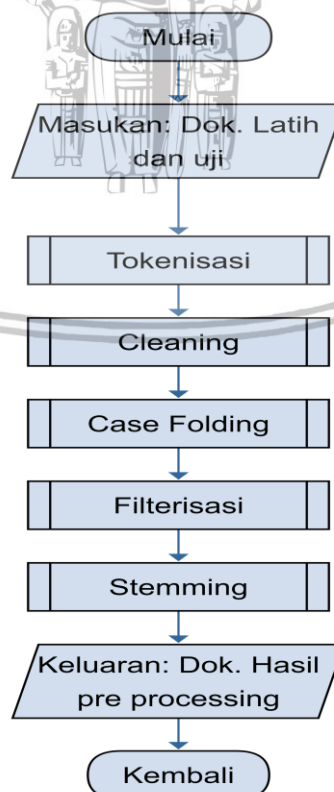
Gambar 4.1 Gambaran Umum Sistem

Gambar 4.1, melakukan input data latih dan data uji. Data latih tersebut akan melakukan proses tahap *Pre Processing* teks dengan tokenisasi, data *cleaning*, *case folding*, *filterisasi*, dan *stemming*. Hasil dari tahapan *Pre Processing* tersebut tahap berikutnya yaitu menghitung pembobotan kata yang meliputi *Term Frequency-Inverse Document Frequency* (TF-IDF) selanjutnya melakukan tahap proses klasifikasi teks menggunakan metode *Support Vector Machine* dengan memakai kernel *Polynomial kernel degree 2*. Pada tahap akhir melakukan pembobotan jumlah *retweet* (non-tekstual).

Selanjutnya untuk melakukan pembobotan pada jumlah *retweet* maka terlebih dahulu menghitung nilai normalisasi *min-max* di dokumen *tweet*. Bobot diberikan sesuai dengan jumlah *retweet* yang terdapat pada dokumen untuk membandingkan jumlah *retweet* pada data latih dan data uji. Selanjutnya hasil bobot tekstual digabungkan dengan bobot jumlah *retweet* (non-tekstual). Melalui hasil dari penggabungan tersebut akan memperoleh suatu nilai, sehingga mampu mengetahui dokumen yang bernilai positif atau negatif setelah dilakukan proses klasifikasi.

4.2 Pre Processing Teks

Gambar 4.2 merupakan diagram alir dari proses *Pre Processing*. Pada tahap *Pre Processing* akan dilakukan proses tokenisasi, data *cleaning*, *case folding*, *filterisasi*, dan *stemming*.

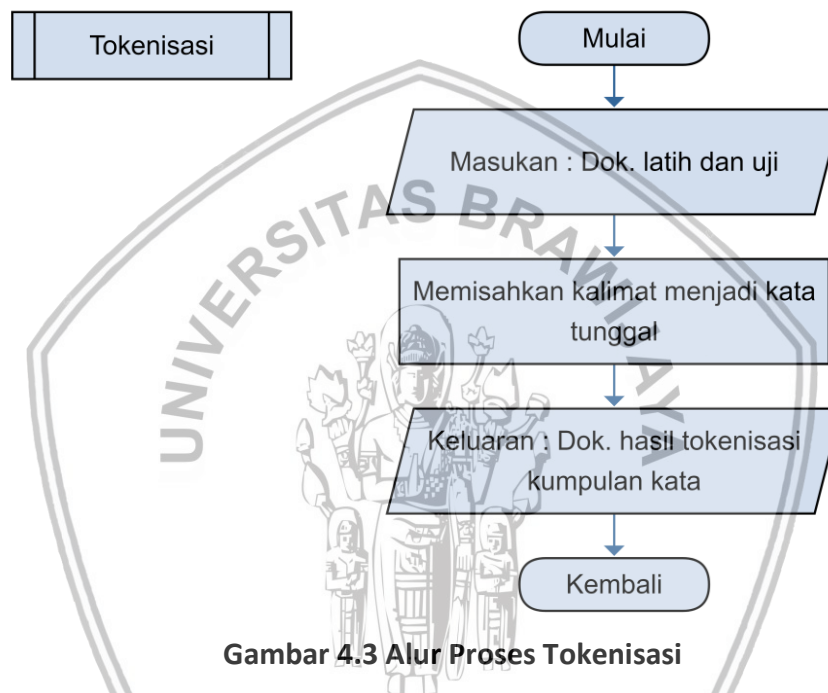


Gambar 4.2 Alur Proses *Pre Processing* Teks

Proses tokenisasi dan cleaning akan memecah setiap kata menjadi token dan menghilangkan *username* (@), *link* (http), *hashtags* (#) serta *retweet*. *Case folding* mengubah huruf besar menjadi huruf kecil. Tahap *filterisasi* akan menggunakan dokumen *stoplist.txt* yang berfungsi untuk menghapus kata yang terdapat pada *stoplist*. Untuk *stemming* akan menggunakan *library JSastrawi*.

4.2.1 Tokenisasi

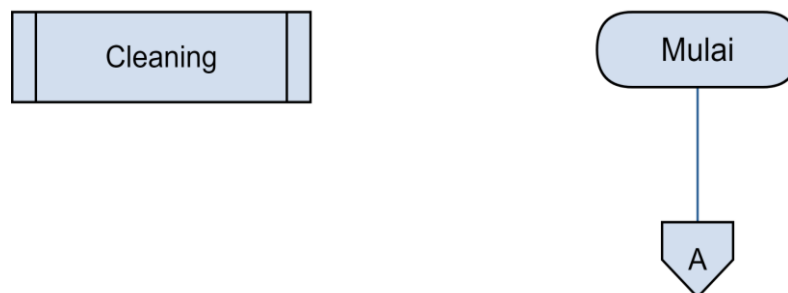
Tokenisasi merupakan langkah awal yang memiliki fungsi untuk memecah tiap kata pada suatu dokumen menjadi kata tunggal. Sehingga seluruh kata pada kalimat dapat dipisahkan. Gambar 4.3 merupakan alur proses tokenisasi.

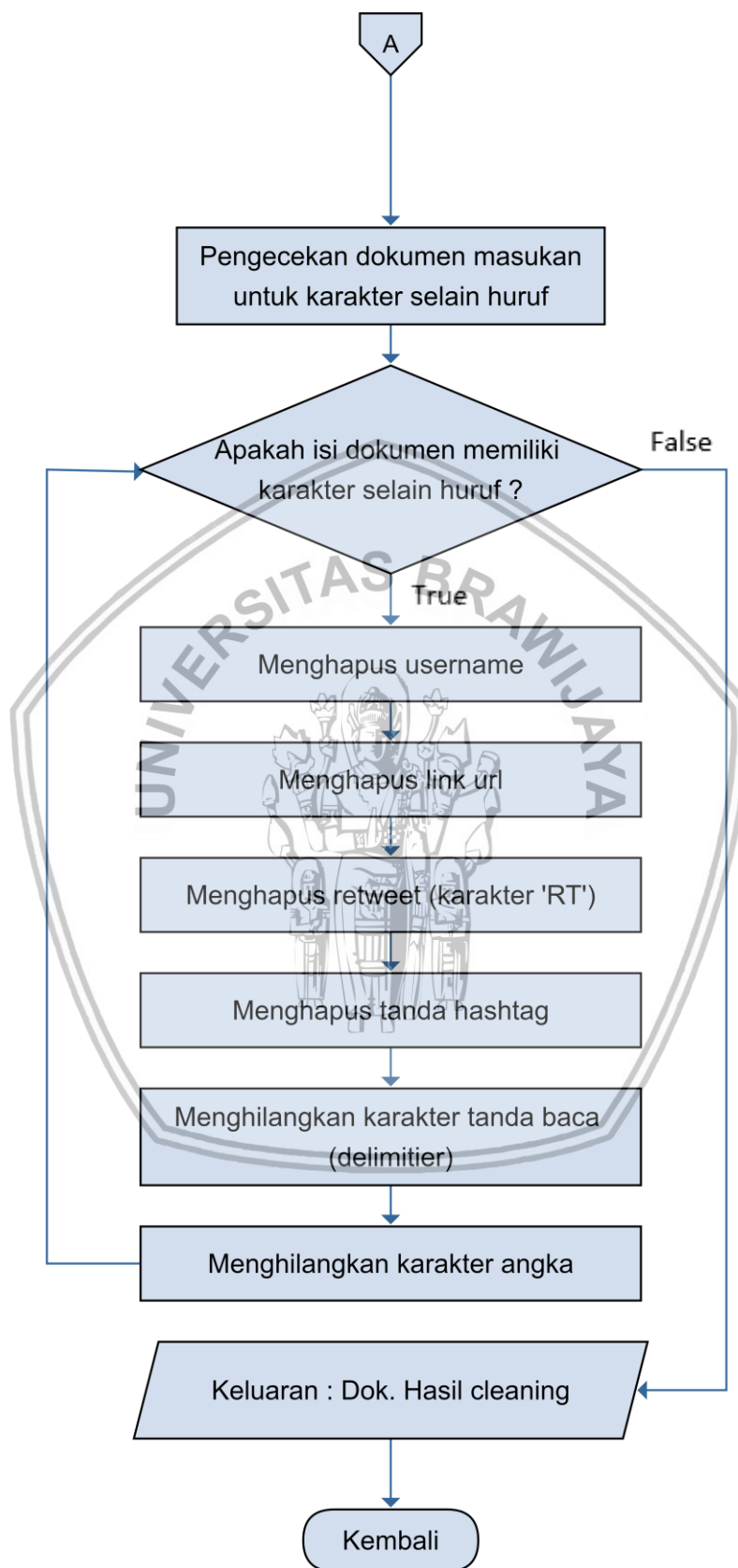


Gambar 4.3 Alur Proses Tokenisasi

4.2.2 Cleaning

Cleaning merupakan tahap kedua dalam *Pre Processing* teks memiliki fungsi untuk membersihkan setiap kata pada suatu dokumen yang tidak memiliki makna dalam pemrosesan teks seperti *username* (@), *link url* (http), *hashtag* (#), *Retweet* (RT) dan menghapus semua karakter selain huruf. Kemudian tanda baca yang terdapat dalam dokumen. Gambar 4.4 merupakan alur proses *Cleaning*.

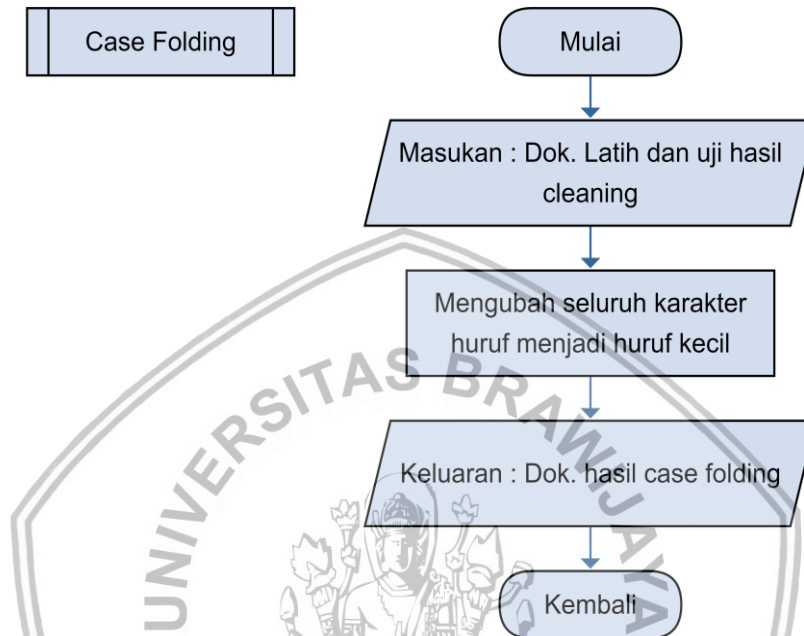




Gambar 4.4 Alur Proses Cleaning

4.2.3 Case Folding

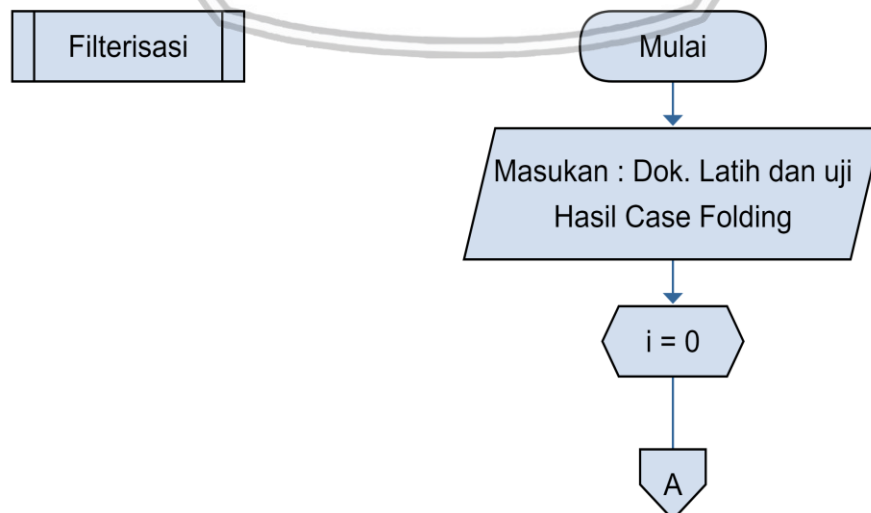
Case folding merupakan tahap ketiga dalam *Pre Processing*. *Case folding* bertujuan untuk mengubah semua karakter pada setiap kata yang terdapat huruf kapital menjadi *lowercase* atau huruf kecil dalam suatu dokumen. Gambar 4.5 merupakan alur proses *case folding*.

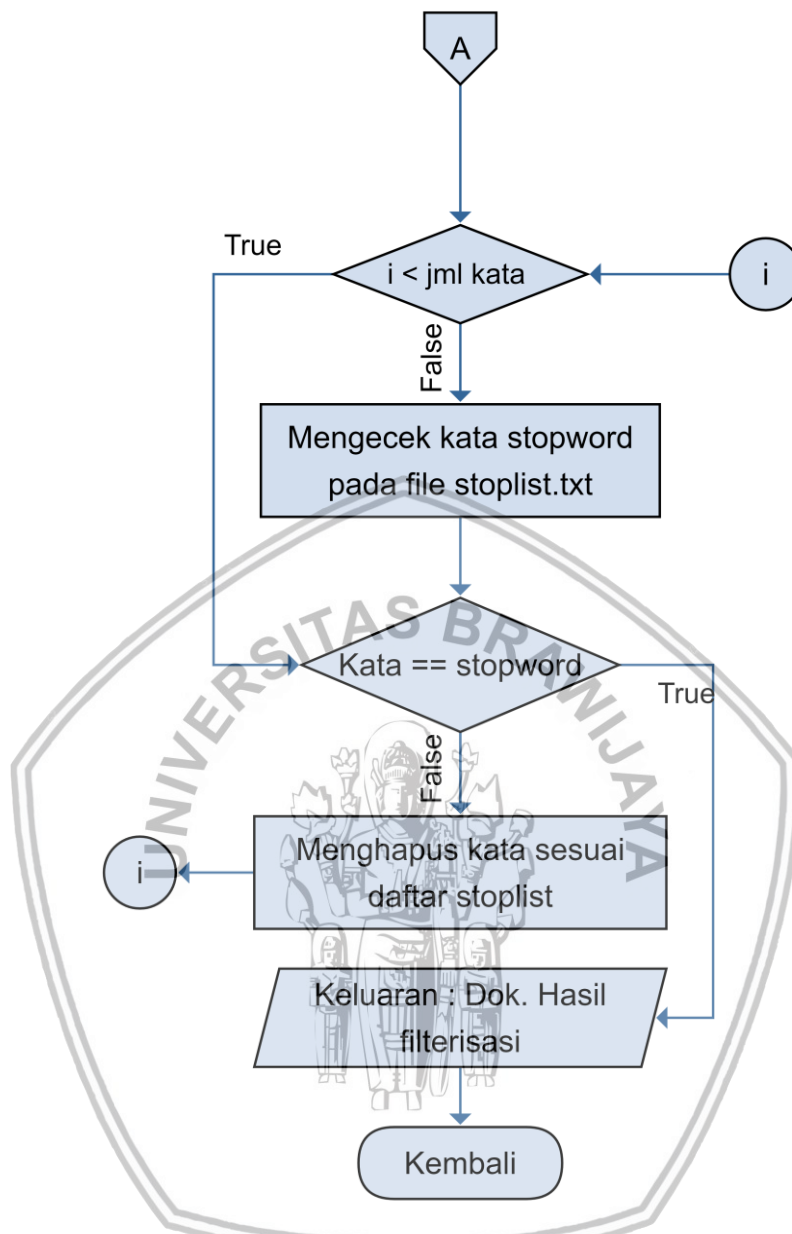


Gambar 4.5 Alur Proses *Case Folding*

4.2.4 Filterisasi

Filterisasi merupakan tahap ke empat dalam *Pre Processin*. Saat proses *case folding* sudah dilakukan sebelumnya. Gambar 4.6 merupakan alur proses *filterisasi*.



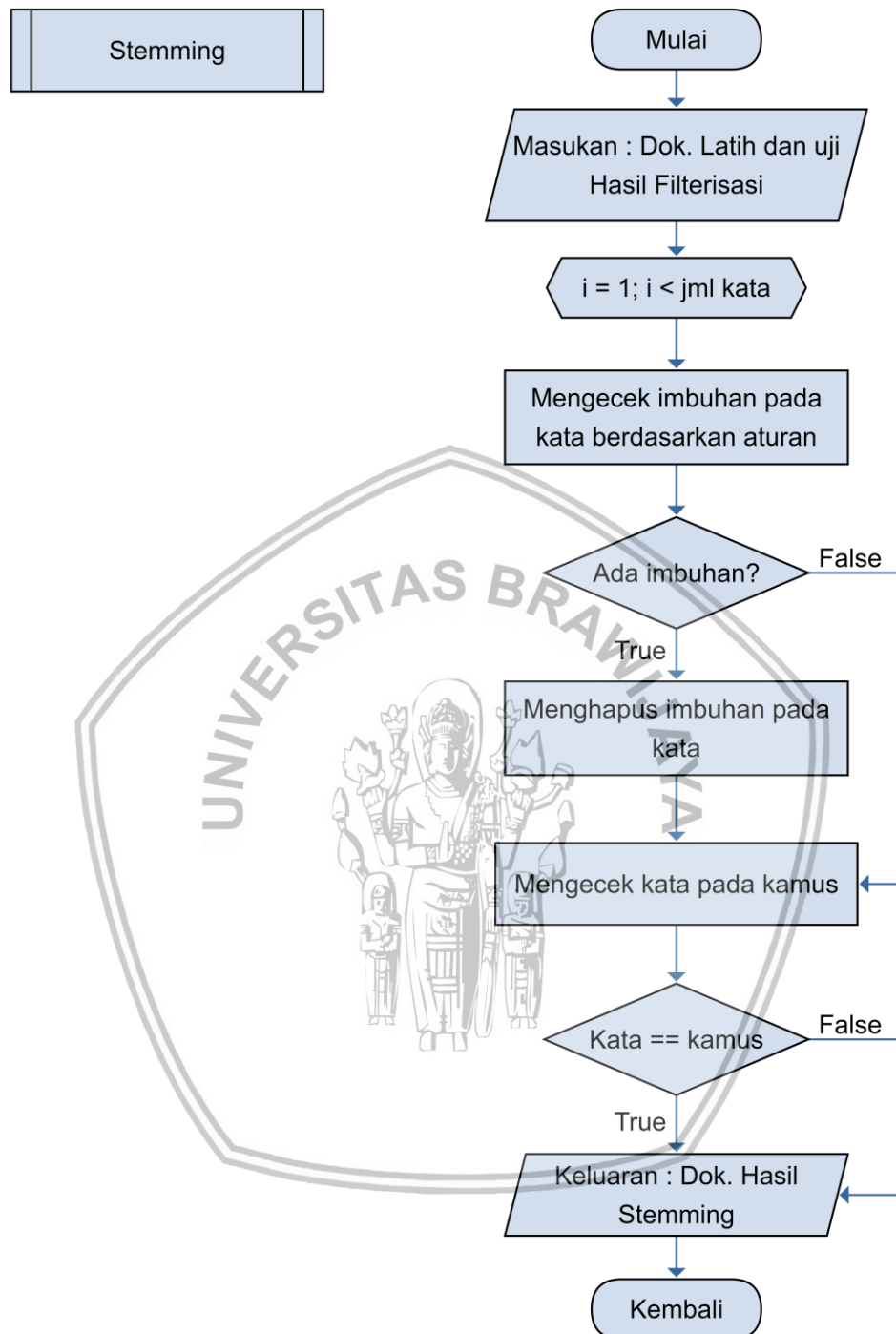


Gambar 4.6 Alur Proses *Filterisasi*

Proses ini bertujuan untuk menghapuskan kata yang tidak memberikan pengaruh pada proses klasifikasi *opinion mining*. Proses filterisasi dilakukan ketika masukkan data dari hasil proses *case folding* dilakukan pengecekan pada tiap kata di suatu dokumen, jika terdapat pada daftar *stoplist*, maka kata tersebut akan dihapus. *Stoplist* merupakan daftar kumpulan kata-kata dalam kamus yang tidak diperlukan untuk proses perhitungan dan akan mengurangi akurasi jika tetap digunakan.

4.2.5 Stemming

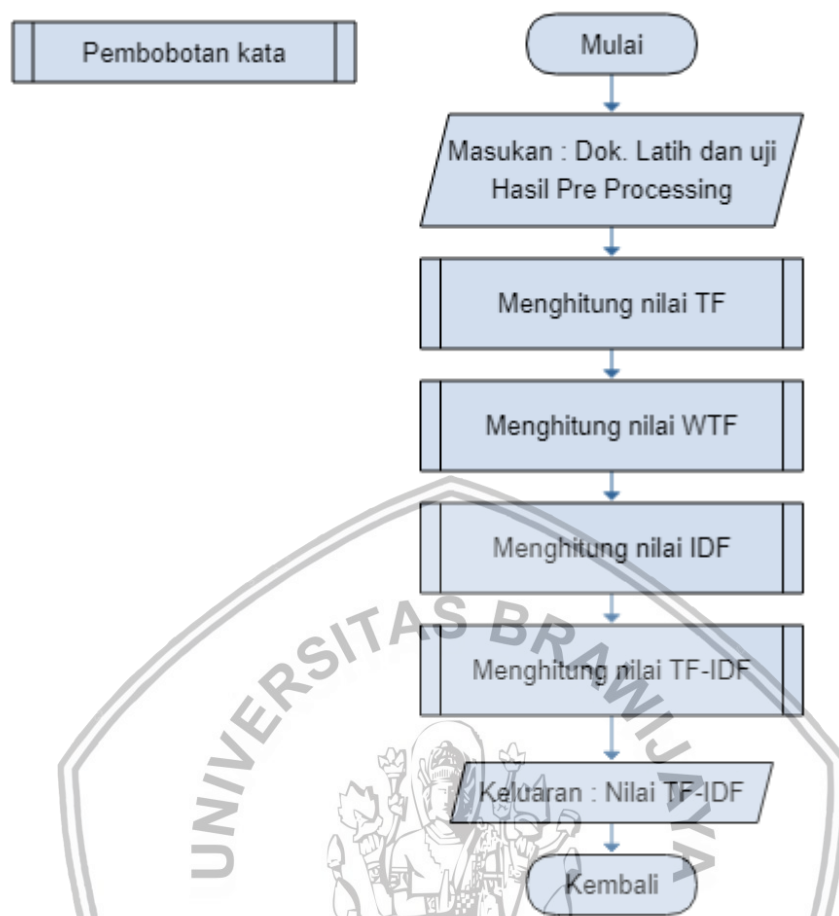
Stemming merupakan tahap akhir dalam *Pre Processing* teks dari hasil proses *filterisasi*. Proses ini bertujuan untuk mengubah kata berimbuhan menjadi suatu kata dasar dalam suatu dokumen. Gambar 4.7 merupakan alur proses *stemming*.



Gambar 4.7 Alur Proses *Stemming*

4.3 Pembobotan Kata

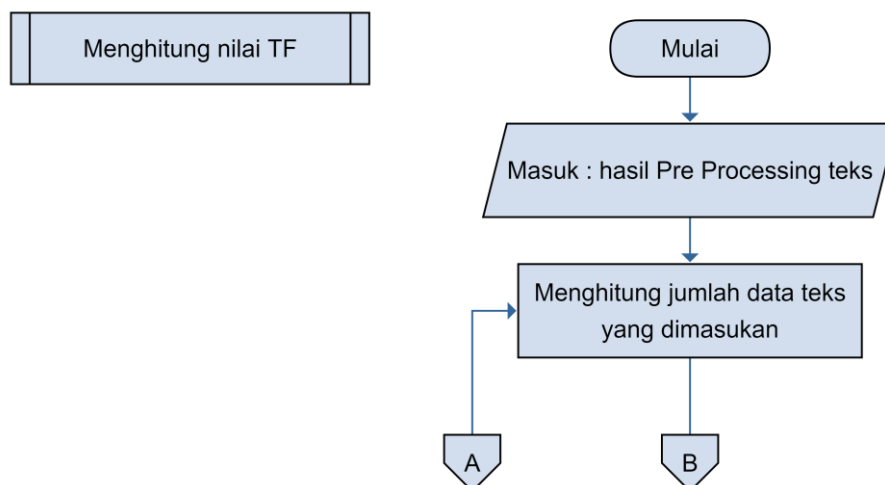
Pembobotan kata merupakan sebuah proses ketika tahapan *Pre Processing* selesai dikerjakan. Tahap awal melakukan proses ini dimulai dengan menerima masukan berupa hasil dari tahapan *Pre Processing* teks yang selanjutnya melakukan proses perhitungan TF, W_{tf} , IDF, dan TF-IDF. Gambar 4.8 merupakan alur proses pembobotan kata.

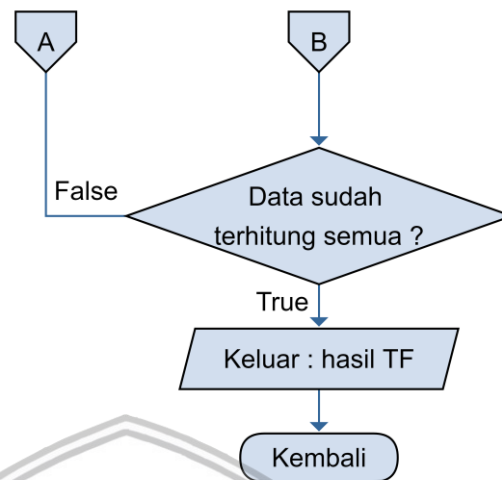


Gambar 4.8 Alur Proses Pembobotan Kata

4.3.1 Menghitung Nilai TF

TF merupakan tahapan awal yang dilakukan dalam proses pembobotan kata. Proses dimulai saat menerima masukan dari hasil yang diperoleh pada *Pre Processing* teks, selanjutnya menghitung jumlah kemunculan kata pada tiap dokumen. Gambar 4.9 merupakan alur proses perhitungan nilai TF.

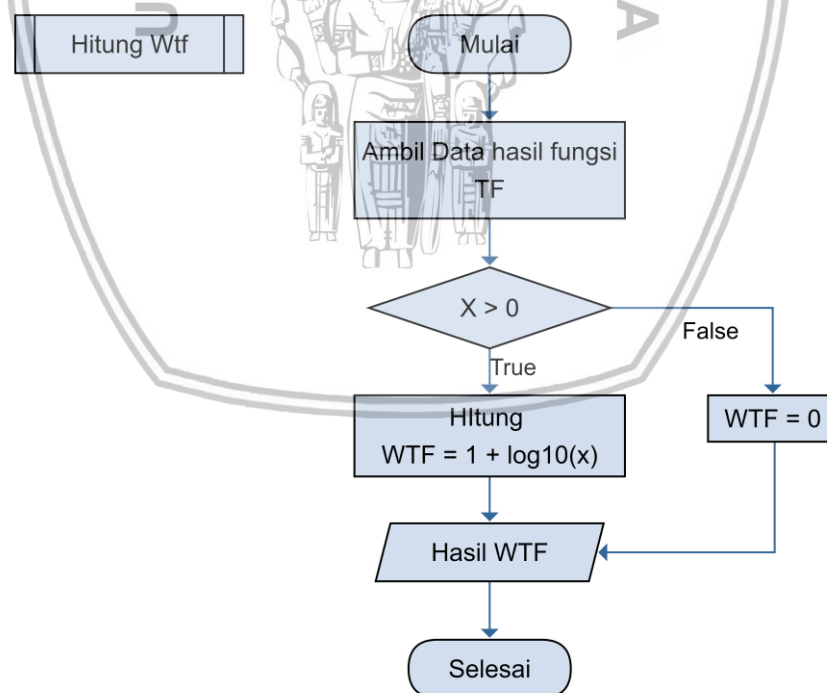




Gambar 4.9 Alur Proses Perhitungan Nilai TF

4.3.2 Menghitung Nilai WTF

WTF merupakan proses tahapan kedua yang dilakukan dalam proses pembobotan kata. Melakukan proses perhitungan nilai TF sesuai dengan rumus perhitungan algoritme. Gambar 4.10 merupakan alur proses perhitungan nilai TF.

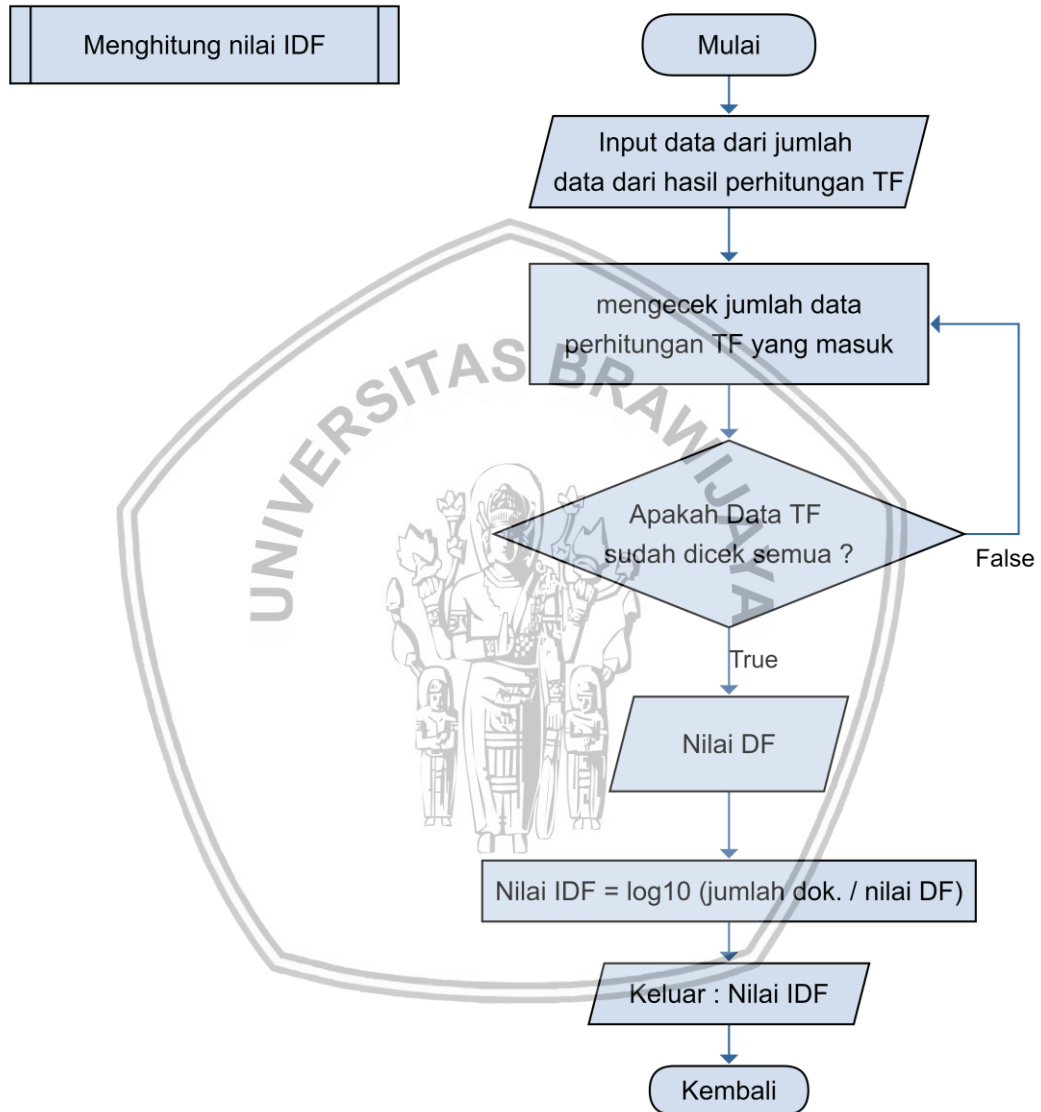


Gambar 4.10 Proses Perhitungan Nilai WTF

4.3.3 Menghitung Nilai IDF

Perhitungan nilai IDF merupakan proses tahapan ketiga yang dilakukan setelah selesai melakukan perhitungan nilai TF dan WTF. IDF menyatakan banyaknya kata-

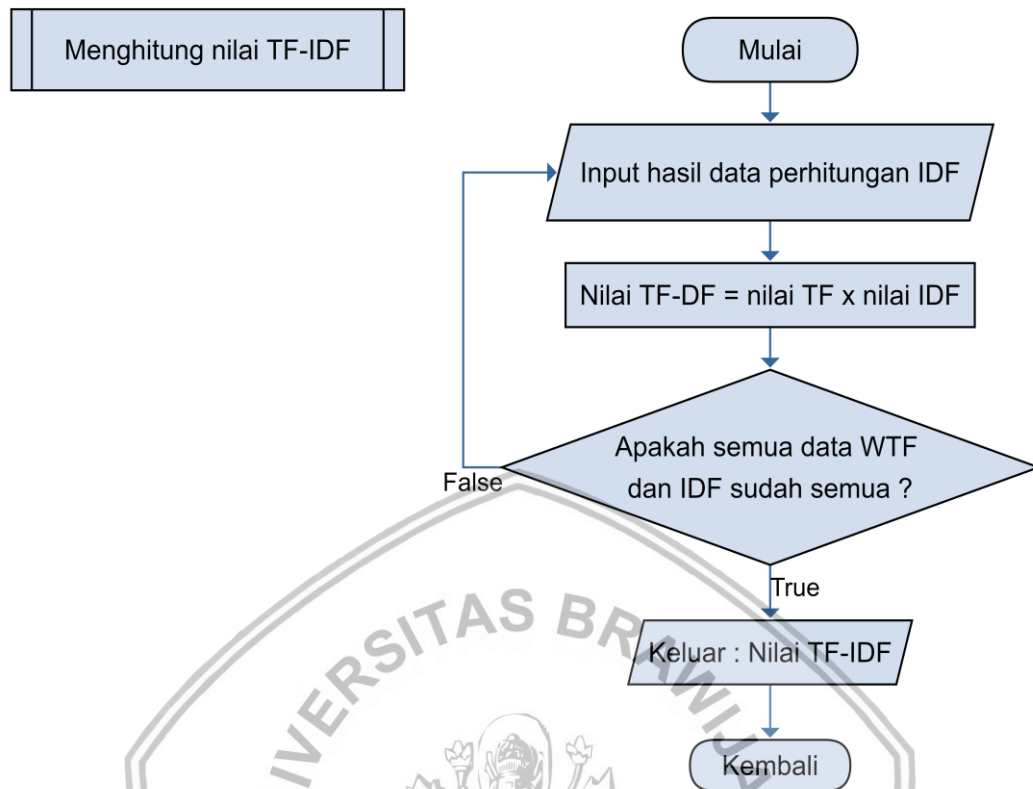
terdapat di dokumen. IDF yaitu nilai bobot suatu *term* yang dihitung dari seringnya kata yang muncul dibagi dengan *df*. Nilai *df* merupakan jumlah yang terdapat *term* t di dalam banyak dokumen. Proses ini mulanya menerima masukan berupa hasil dari proses menghitung nilai TF dan WTF. Proses ini akan berulang sampai semua dokumen selesai melakukan proses perhitungan. Gambar 4.11 alur proses perhitungan nilai IDF.



Gambar 4.11 Alur Proses Perhitungan Nilai IDF

4.3.4 Menghitung TF-IDF

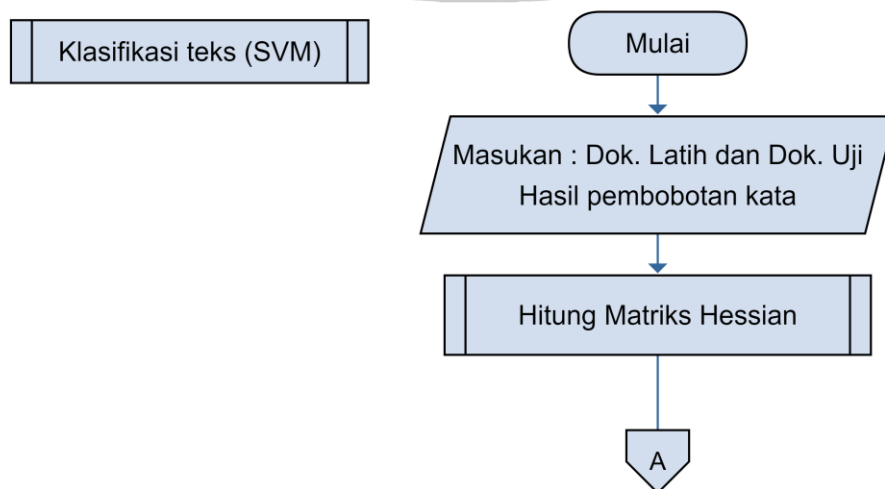
Perhitungan nilai TF-IDF merupakan proses tahapan akhir yang dilakukan setelah melakukan proses perhitungan nilai IDF. TF-IDF bertujuan untuk menghitung bobot nilai TF dan IDF dalam suatu dokumen pada tiap kata. Proses dimulai saat mendapatkan hasil dari nilai IDF, kemudian akan menghitung jumlah nilai TF-IDF, proses akan berulang hingga proses perhitungan selesai dilakukan dalam dokumen. Gambar 4.12 merupakan alur proses perhitungan nilai TF-IDF.

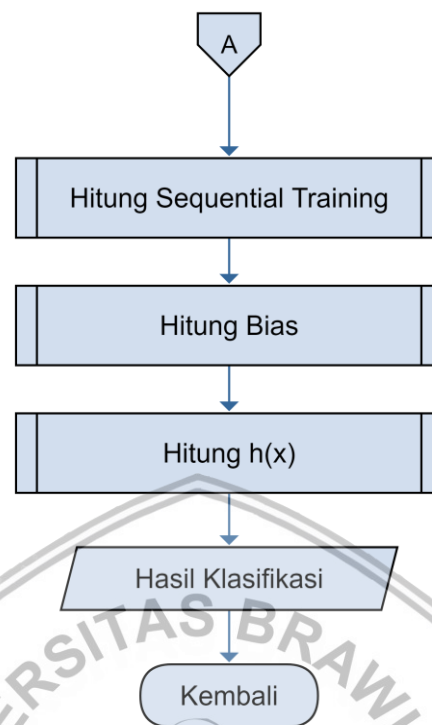


Gambar 4.12 Alur Proses Perhitungan Nilai TF-IDF

4.4 Klasifikasi teks *Support Vector Machine*

Klasifikasi teks *Support Vector Machine* suatu tahapan yang dilakukan setelah proses pembobotan teks selesai dilakukan. Beberapa proses yang harus dilakukan yaitu melakukan perhitungan dengan menggunakan metode *Support Vector Machine*. Tahap-tahap yang dilakukan yaitu hitung kernel *polynomial*, Hitung matriks hessian, *Sequential Training*, Hitung Bias, dan Hitung $h(x)$. Hasil yang akan didapatkan yaitu kelas positif atau negatif. Gambar 4.13 merupakan alur proses perhitungan klasifikasi teks *Support Vector Machine*.

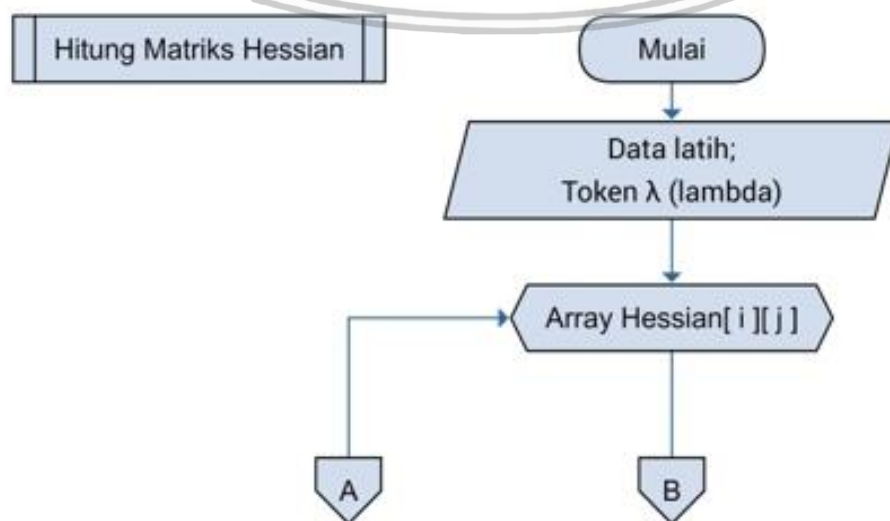


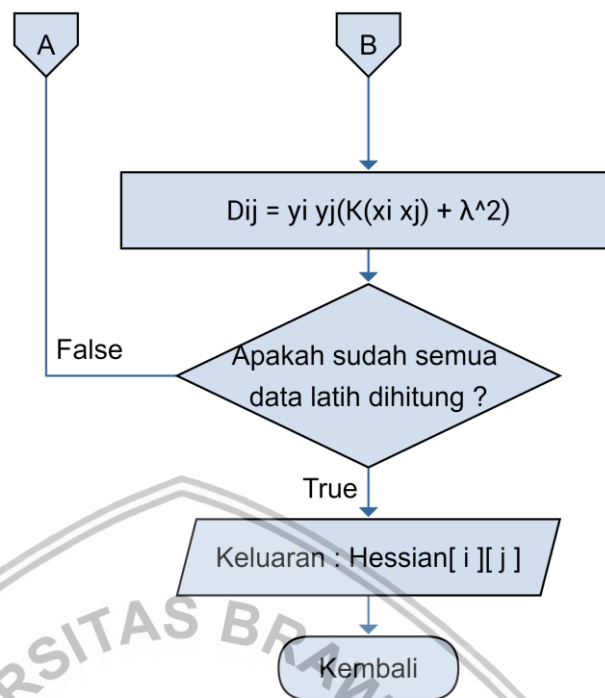


Gambar 4.13 Alur Proses Klasifikasi *Support Vector Machine*

4.4.1 Menghitung Matriks Hessian

Perhitungan matriks hessian merupakan tahapan awal yang dilakukan dalam proses klasifikasi teks menggunakan metode *Support Vector Machine*. Langkah perhitungan matriks hessian dilakukan dengan Persamaan 2.6. Kernel *polynomial* berderajat merupakan kernel digunakan pada penelitian ini. Kernel didapatkan melalui hasil perhitungan antara dua dokumen untuk diproses, matriks hessian akan digunakan pada perhitungan iterasi berikutnya. Gambar 4.14 merupakan alur proses perhitungan matriks hessian.

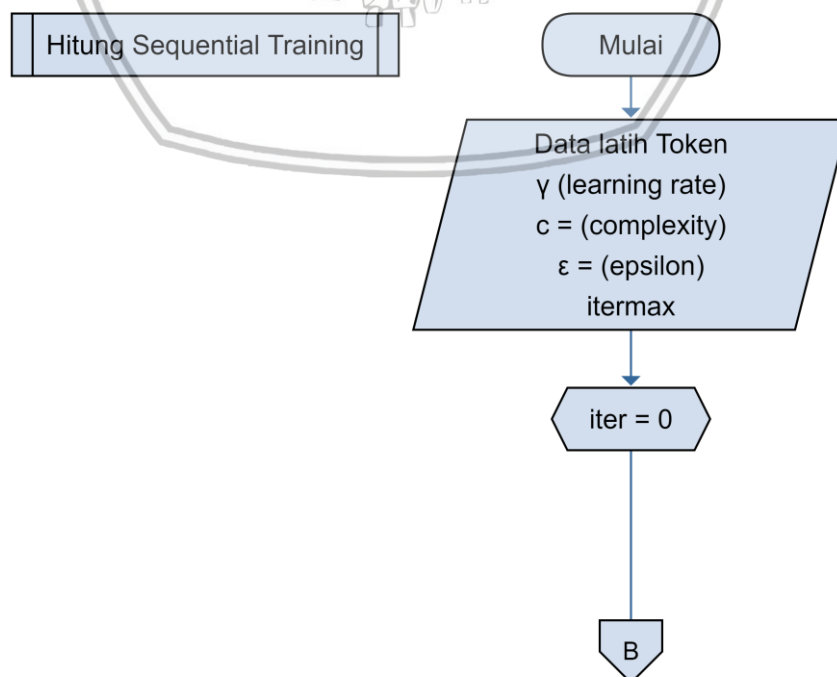


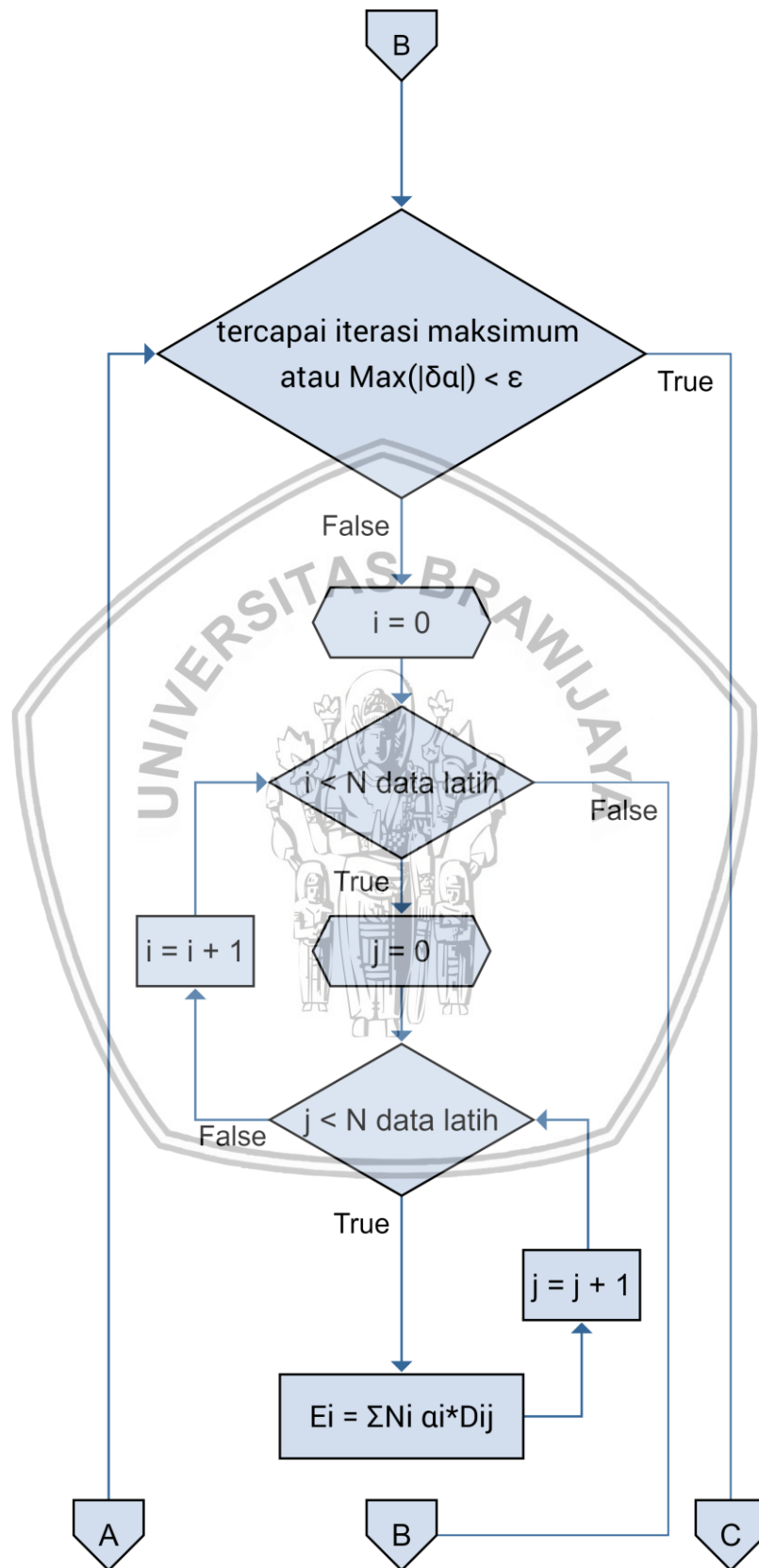


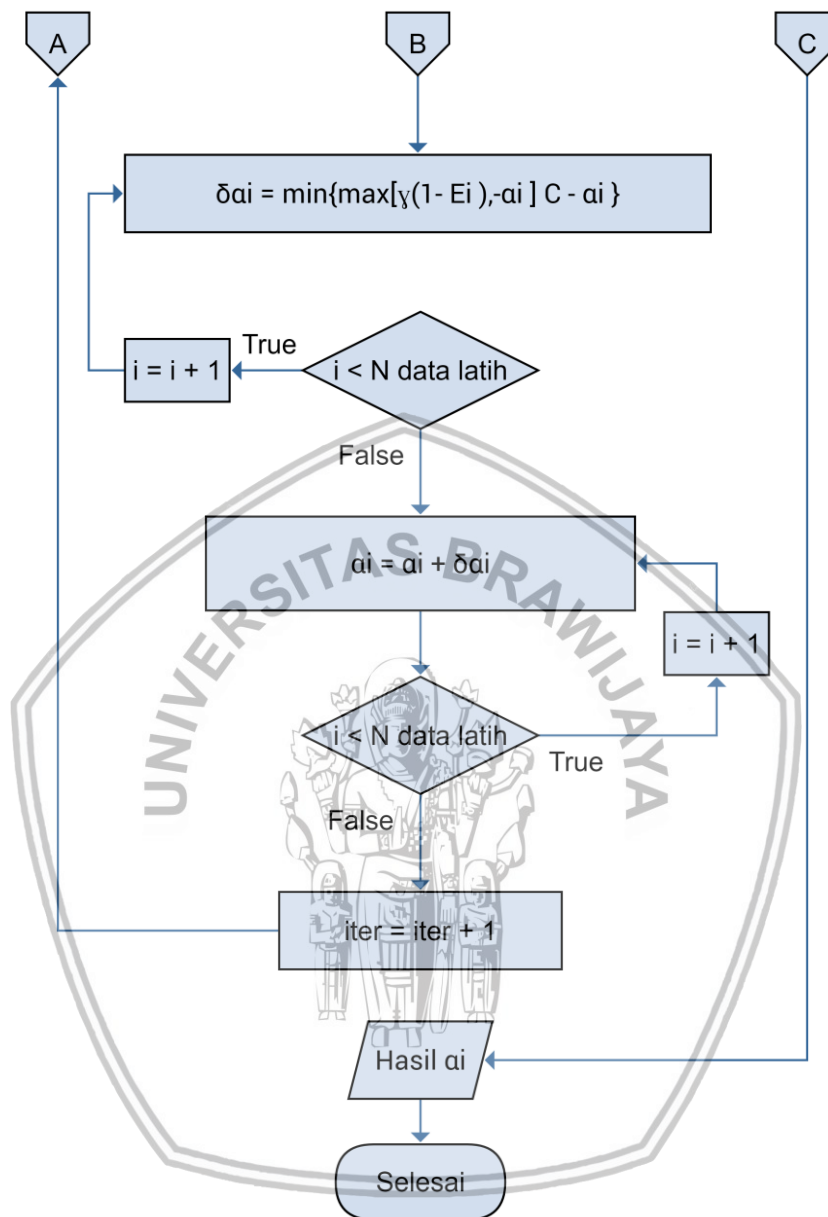
Gambar 4.14 Alur Proses Hitung Matriks Hessian

4.4.2 Sequential Training

Tahap ini berfungsi untuk proses menghitung nilai E_i , $\delta\alpha_i$ dan α_i . Nilai E_i didapatkan melalui total jumlah perkalian matriks Hessian pada Persamaan 2.7. Sementara itu nilai $\delta\alpha_i$ didapatkan pada Persamaan 2.8. dan nilai α_i didapatkan dari Persamaan 2.9. Nilai α_i baru dihitung dari nilai α_i lama ditambah dengan nilai $\delta\alpha_i$. Gambar 4.15 merupakan alur proses *sequential training*.



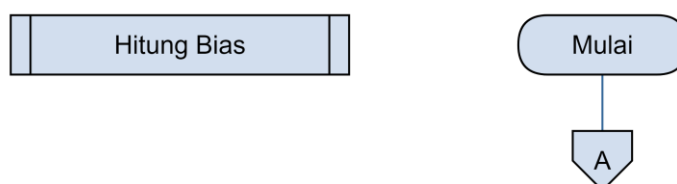


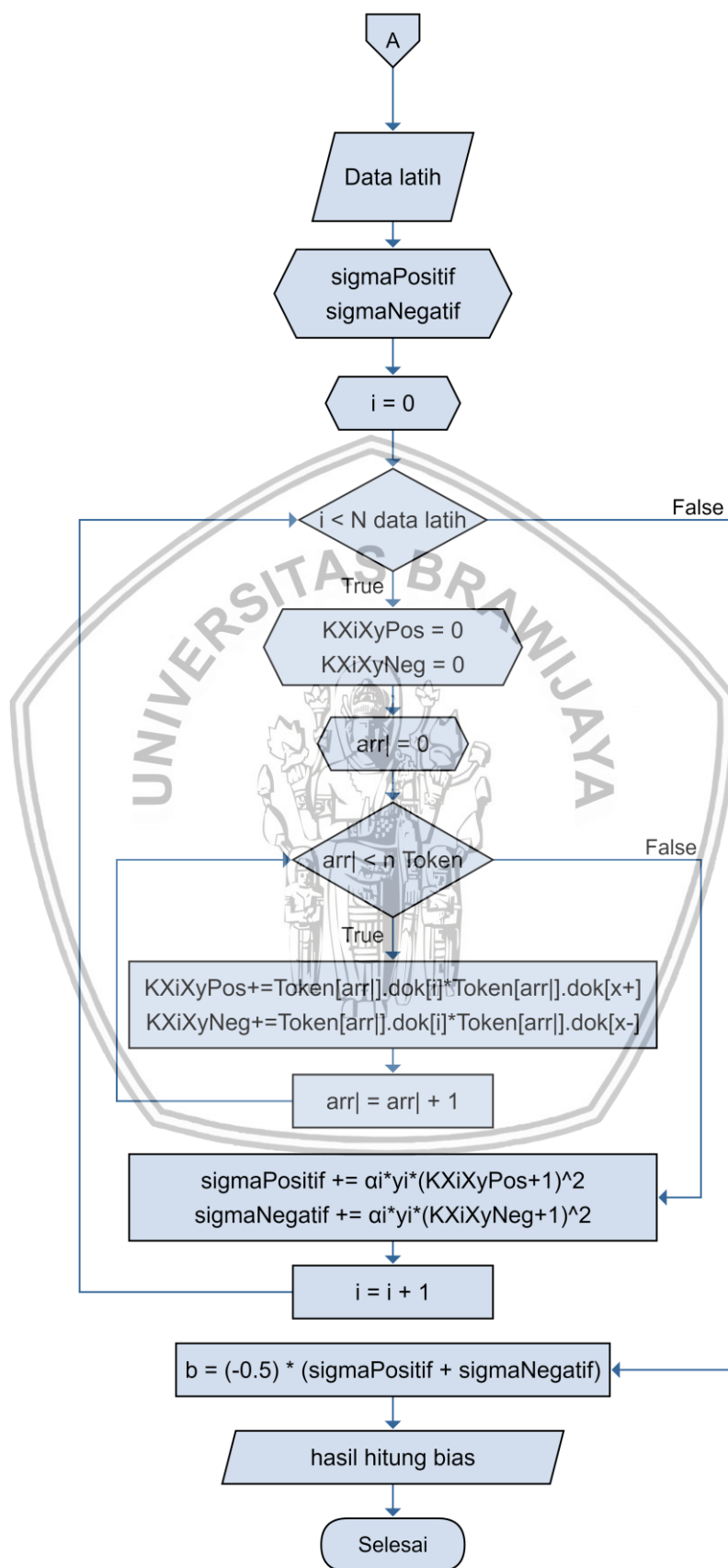


Gambar 4.15 Alur Proses *Sequential Training*

4.4.3 Menghitung Bias

Perhitungan bias merupakan tahap untuk mendapatkan nilai bias yang terdapat di Persamaan 2.10. Gambar 4.16 merupakan alur proses perhitungan bias.

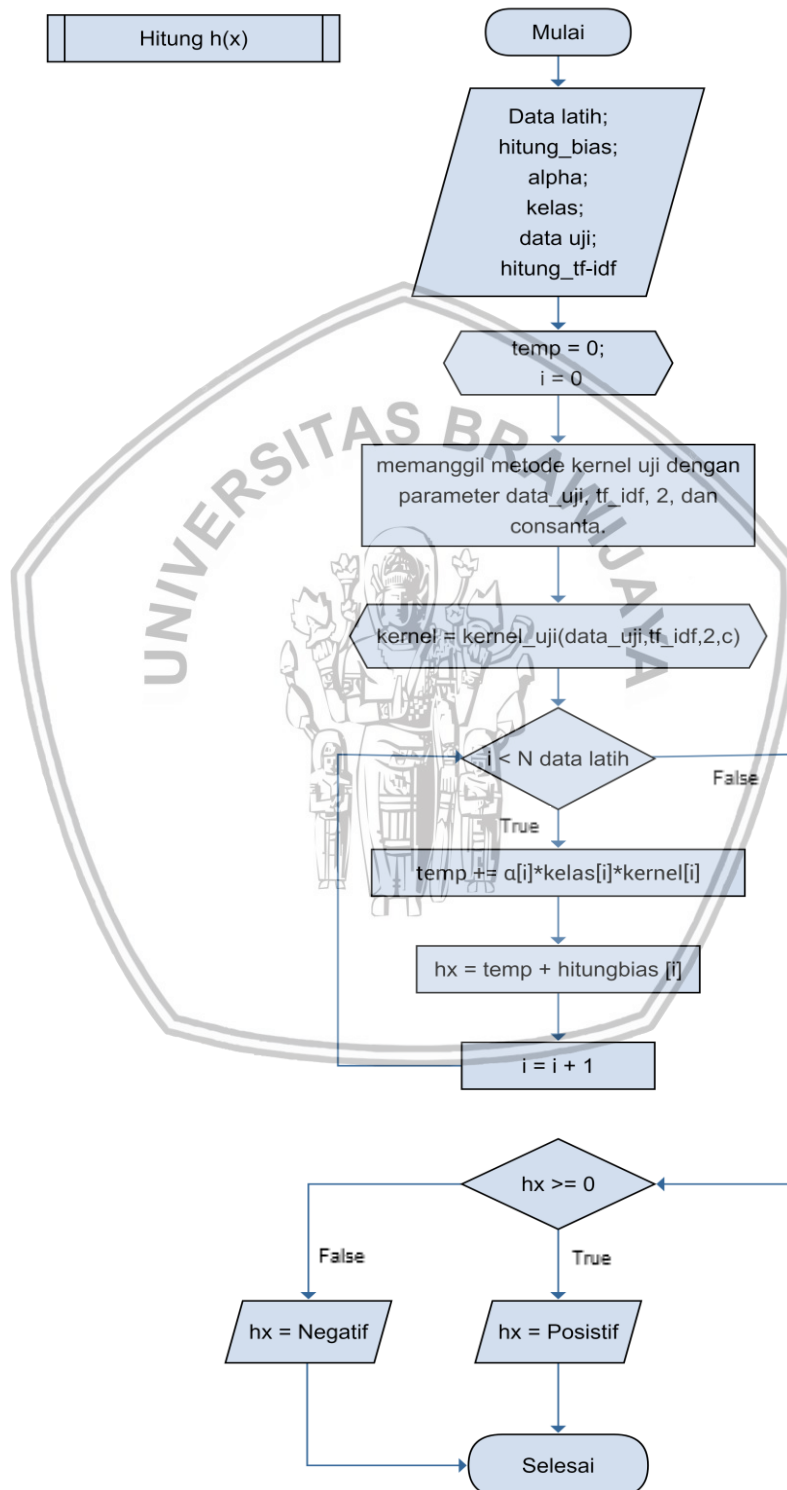




Gambar 4.16 Alur Proses Perhitungan Bias

4.4.4 Menghitung $h(x)$

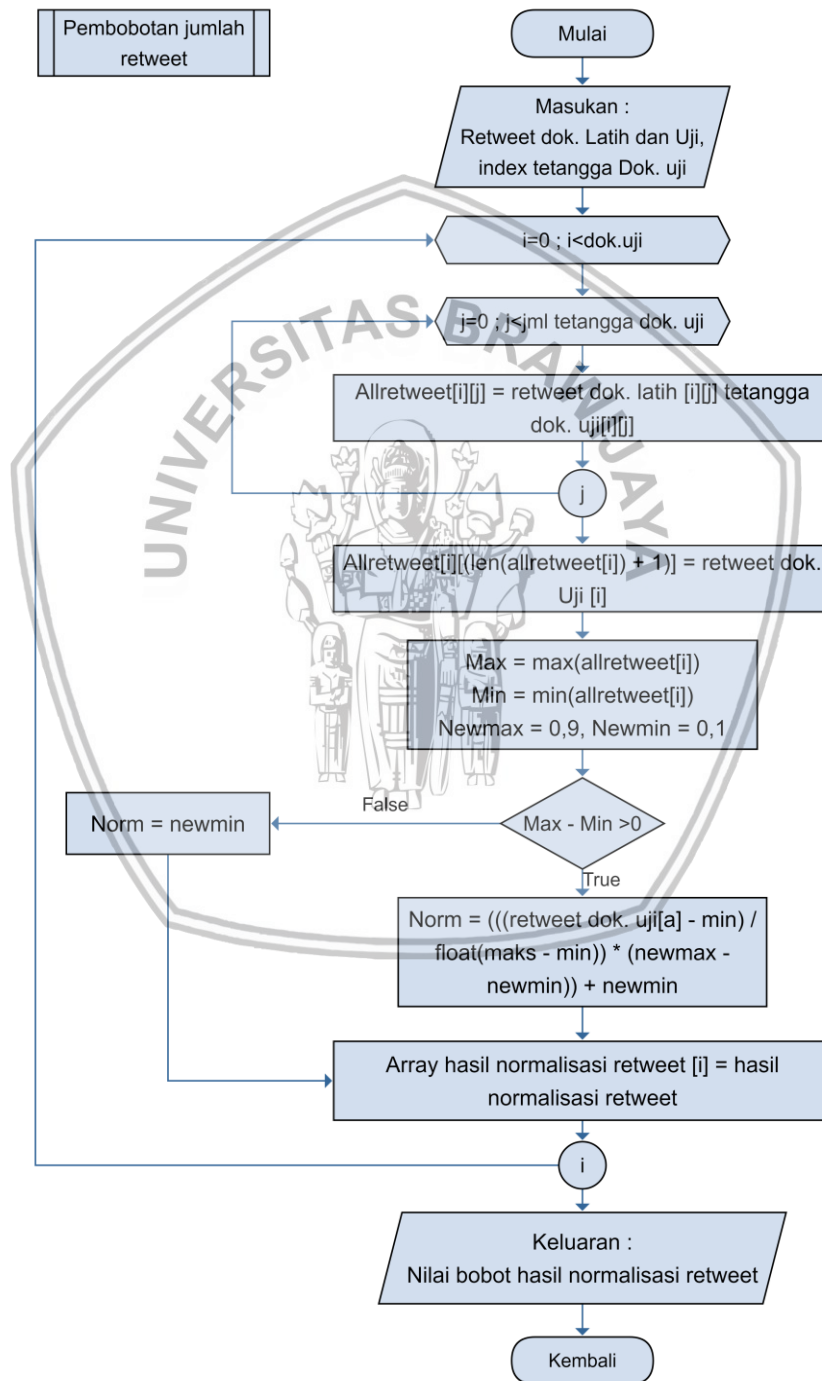
Hitung kemiripan merupakan tahap untuk melakukan perhitungan nilai pada Persamaan 2.11 antara data uji dengan data latih. Gambar 4.17 merupakan alur proses perhitungan kemiripan.



Gambar 4.17 Alur Proses Hitung $h(x)$

4.5 Pembobotan Jumlah *Retweet* (non-tekstual)

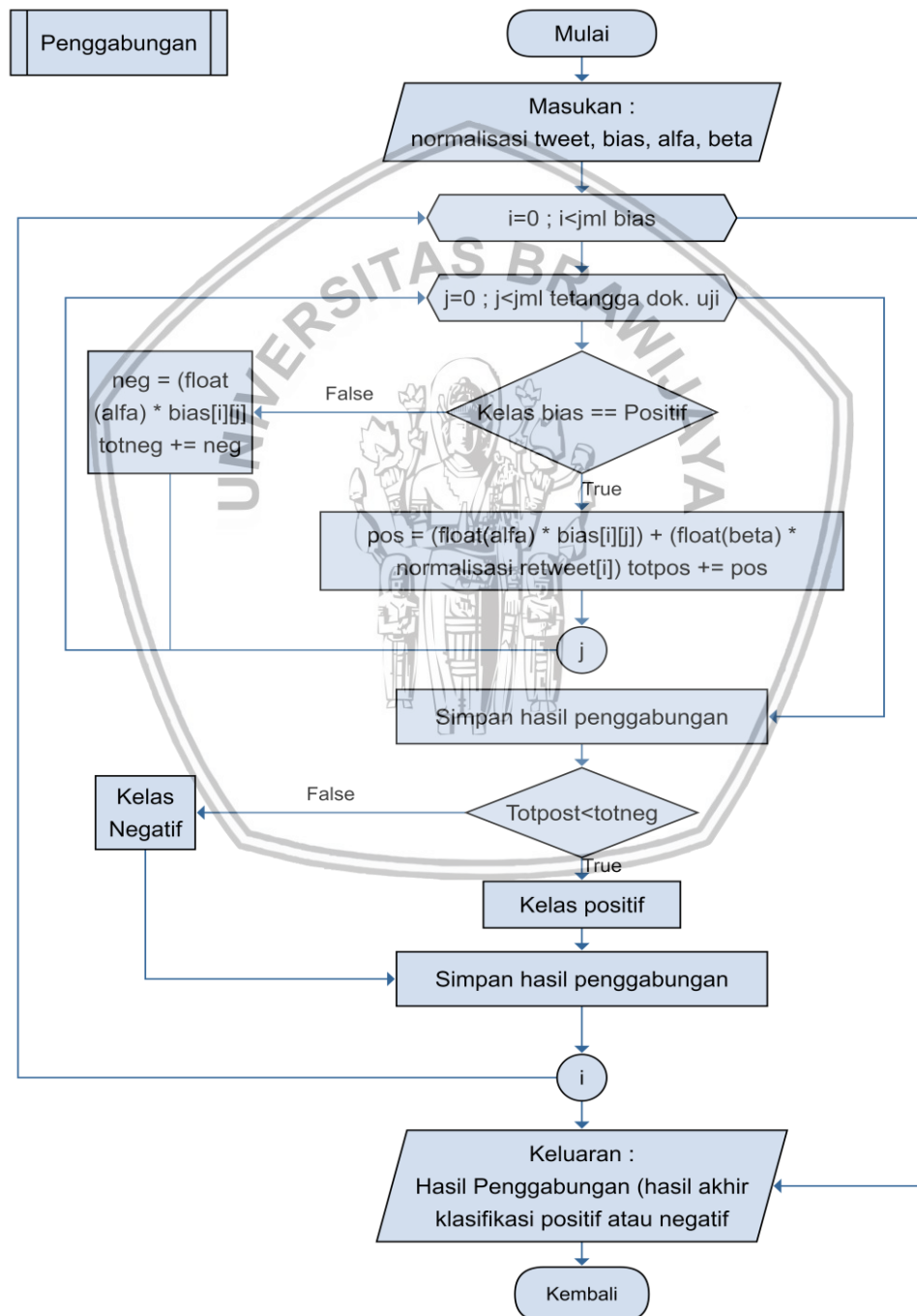
Pada bagian ini bertujuan untuk melakukan proses klasifikasi non-tekstual yang merupakan fitur tambahan dalam melakukan proses klasifikasi. Pemberian bobot diberikan sesuai dengan jumlah *retweet* pada dokumen *tweet*, selanjutnya dilakukan normalisasi perhitungan Min-max. Gambar 4.18 merupakan alur proses perhitungan pembobotan jumlah *retweet*.



Gambar 4.18 Pembobotan jumlah *Retweet* (non-tekstual)

4.6 Penggabungan

Bagian ini bertujuan untuk melakukan proses penggabungan bobot tekstual pada nilai yang didapatkan pada klasifikasi teks *Support Vector Machine* dan pembobotan non-tekstual. Selanjutnya melakukan proses perhitungan yang terdapat pada Persamaan 2.12. Penambahan konstanta α dan β dijadikan konstanta pembanding antara bobot tekstual dan non tekstual. Nilai α dan β untuk mengetahui tingkat akurasi, apakah penambahan fitur akan memberikan suatu pengaruh atau tidak. Gambar 4.20 merupakan alur proses penggabungan.



Gambar 4.19 Alur Proses Penggabungan

4.7 Manualisasi *Support Vector Machine* dan Pembobotan Jumlah *Retweet*

Tahap ini menguraikan proses perhitungan manualisasi pada percobaan data latih dan data uji. Manualisasi ini merupakan deksripsi umum dalam penelitian ini. *Kernel* yang digunakan yaitu *polynomial* berderajat dua. Beberapa tahapan yang akan dilakukan untuk perhitungan manual yaitu, manualisasi *Pre Processing* teks, pembobotan kata, perhitungan *Support Vector Machine*, perhitungan normalisasi Min-max untuk pembobotan jumlah *retweet* dan kombinasi antara pembobotan kata hasil klasifikasi *Support Vector Machine* (tekstual). Data yang digunakan yaitu 6 dokumen untuk data latih dan 3 dokumen untuk data uji sebagai perhitungan manual. Pada Tabel 4.1 dan Tabel 4.2 menunjukkan sampel untuk data latih dan data uji.

Tabel 4.1 Data Latih tweet

NO	Data latih tweet	Jumlah <i>Retweet</i>	Kategori
1	Diem diem aja bro #imporberas #imporkeprihatinan	0	Negatif
2	Masalah Anda ibarat roda.... Kadang kempes kadang aman. Begitulah hidupmu. Maklumlah kamu hanya sebagai Rakyat. Ikut saja aturan mainnya. #selesai saatnya jalan, jalankan. saatnya bayar yoo bayarlah. #imporberas #pilihan2018	0	Positif
3	Negeri yang maju harus rajin impor, biar kapitalis semakin makmur dan kemiskinan semakin menjamur. Sehingga lambat laun bak sapi mati kelaparan diatas rerumputan. #Importerus #imporberas	0	Positif
4	Dianggap Mengkhianati Petani, MPR-DPR Desak Jokowi Hentikan Impor Beras https://goo.gl/fb/NAMoJ1 #ekonomi #imporberas	0	Negatif
5	BI Dukung Kran Impor Beras ke Batam Dibuka #imporberas	2	Positif
6	Rizal Ramli: Surya Paloh Dukung Jokowi, Tetapi Menteri Gerogetin Elektabilitas #ImporBeras	4	Positif

Tabel 4.2 Data Uji tweet

NO	Data uji tweet	Jumlah <i>Retweet</i>	Kategori
1	Mainnya ke sawah, tapi beras aja impor @jokowi	10	?
2	Ketum PAN: Stop Impor Beras, Biarkan Petani Nikmati Panen #ImporBeras	3	?
3	Impor beras untuk kesejahteraan rakyat? Bagus tuh... ntar saat pilpres 2019, saya sarankan itu jadi motto baru! Jangan cuma beras, tapi semua komoditi impor @jokowi	0	?

4.7.1 Manualisasi Tokenisasi

Tahap melakukan pemecahan teks berupa kumpulan kalimat menjadi kata-kata. Tabel 4.3 dan Tabel 4.4 menunjukkan proses manualisasi tokenisasi.

Tabel 4.3 Manualisasi Tokenisasi (Data Latih)

NO	Data latih tweet	Hasil data latih tweet Tokenisasi
1	Diem diem aja bro #imporberas #imporkeprihatinan	['Diem', 'diem', 'aja', 'bro', '#imporberas', '#imporkeprihatinan']
2	Masalah Anda ibarat roda.... Kadang kempes kadang aman. Begitulah hidupmu. Maklumlah kamu hanya sebagai Rakyat. Ikut saja aturan mainnya. #selesai saatnya jalan, jalankan. saatnya bayar yoo bayarlah. #imporberas #pilihan2018	['Masalah', 'Anda', 'ibarat', 'roda....', 'Kadang', 'kempes', 'kadang', 'aman.', 'Begitulah', 'hidupmu.', 'Maklumlah', 'kamu', 'hanya', 'sebagai', 'Rakyat.', 'Ikut', 'saja', 'aturan', 'mainnya.', '#selesai', 'saatnya', 'jalan,', 'jalankan.', 'saatnya', 'bayar', 'yoo', 'bayarlah.', '#imporberas', '#pilihan2018']
3	Negeri yang maju harus rajin impor, biar kapitalis semakin makmur dan kemiskinan semakin menjamur. Sehingga lambat laun bak sapi mati kelaparan diatas rerumputan. #Importerus #imporberas	['Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor,', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', '#Importerus', '#imporberas']
4	Dianggap Mengkhianati Petani, MPR-DPR Desak Jokowi Hentikan Impor	['Dianggap', 'Mengkhianati', 'Petani,', 'MPR-DPR', 'Desak', 'Jokowi', 'Hentikan',

	Beras https://goo.gl/fb/NAMoJ1 #ekonomi #imporberas	'Impor', 'Beras', 'https://goo.gl/fb/NAMoJ1', '#ekonomi', #imporberas']
5	BI Dukung Kran Impor Beras ke Batam Dibuka #imporberas	['BI', 'Dukung', 'Kran', 'Impor', 'Beras', 'ke', 'Batam', 'Dibuka', '#imporberas']
6	Rizal Ramli: Surya Paloh Dukung Jokowi, Tetapi Menteriya Gerogotin Elektabilitas #ImporBeras	['Rizal' 'Ramli:', 'Surya', 'Paloh', 'Dukung', 'Jokowi,', 'Tetapi', 'Menteriya', 'Gerogotin', 'Elektabilitas', #ImporBeras']

Tabel 4.4 Manualisasi Tokenisasi (Data Uji)

NO	Data uji tweet	Hasil data uji tweet Tokenisasi
1	mainnya ke sawah, tapi beras aja impor @jokowi	['mainnya', 'ke', 'sawah,', 'tapi', 'beras', 'aja', 'impor', '@jokowi']
2	Ketum PAN: Stop Impor Beras, Biarkan Petani Nikmati Panen #ImporBeras	['Ketum', 'PAN:', 'Stop', 'Impor', 'Beras,', 'Biarkan', 'Petani', 'Nikmati', 'Panen', '#ImporBeras']
3	Impor beras untuk kesejahteraan rakyat? Bagus tuh... ntar saat pilpres 2019, saya sarankan itu jadi motto baru! Jangan cuma beras, tapi semua komoditi impor @jokowi	['Impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat?', 'Bagus', 'tuh...', 'ntar', 'saat', 'pilpres', '2019', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru!', 'Jangan', 'Cuma', 'beras,', 'tapi', 'semua', 'komoditi', 'impor', '@jokowi']

4.7.2 Manualisasi Cleaning

Pada manualisasi *cleaning* melakukan proses untuk membersihkan setiap kata pada suatu dokumen yang tidak mempunyai pengaruh seperti *username* (@), *link url* (http), *hashtag* (#), *Retweet* (RT) dan menghilangkan seluruh karakter selain huruf dan tanda baca yang terdapat dalam dokumen. Pada Tabel 4.5 dan Tabel 4.6. menunjukkan manualisasi *Cleaning*.

Tabel 4.5 Manualisasi Cleaning (Data Latih)

NO	Data latih tweet	Hasil data latih tweet Cleaning
1	['Diem', 'diem', 'aja', 'bro', #imporberas', '#imporkeprihatinan']	['Diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']
2	['Masalah', 'Anda', 'ibarat', 'roda....', 'Kadang', 'kempes', 'kadang', 'aman.', 'Begitulah', 'hidupmu.', 'Maklumlah', 'kamu', 'hanya', 'sebagai', 'Rakyat.',	['Masalah', 'Anda', 'ibarat', 'roda', 'Kadang', 'kempes', 'kadang', 'aman', 'Begitulah', 'hidupmu', 'Maklumlah', 'kamu', 'hanya', 'sebagai', 'Rakyat',

	'Ikut', 'saja', 'aturan', 'mainnya.', '#selesai', 'saatnya', 'jalan,', 'jalankan.', 'saatnya', 'bayar', 'yoo', 'bayarlah.', '#imporberas', '#pilihan2018']	'Ikut', 'saja', 'aturan', 'mainnya.', 'selesai', 'saatnya', 'jalan', 'jalankan', 'saatnya', 'bayar', 'yoo', 'bayarlah', 'imporberas', 'pilihan2018']
3	['Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor,', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur.', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan.', '#Importerus', '#imporberas']	['Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas']
4	['Dianggap', 'Mengkhianati', 'Petani', 'MPR-DPR', 'Desak', 'Jokowi', 'Hentikan', 'Impor', 'Beras', 'https://goo.gl/fb/NAMoJ1', '#ekonomi', '#imporberas']	['Dianggap', 'Mengkhianati', 'Petani', 'MPR-DPR', 'Desak', 'Jokowi', 'Hentikan', 'Impor', 'Beras', 'ekonomi', 'imporberas']
5	['BI', 'Dukung', 'Kran', 'Impor', 'Beras', 'ke', 'Batam', 'Dibuka', '#imporberas']	['BI', 'Dukung', 'Kran', 'Impor', 'Beras', 'ke', 'Batam', 'Dibuka', 'imporberas']
6	['Rizal', 'Ramli:', 'Surya', 'Paloh', 'Dukung', 'Jokowi,', 'Tetapi', 'Menterinya', 'Gerogotin', 'Elektabilitas', '#ImporBeras']	['Rizal', 'Ramli', 'Surya', 'Paloh', 'Dukung', 'Jokowi', 'Tetapi', 'Menterinya', 'Gerogotin', 'Elektabilitas', 'ImporBeras']

Tabel 4.6 Manualisasi *Cleaning* (Data Uji)

NO	Data uji tweet	Hasil data uji tweet <i>Cleaning</i>
1	['mainnya', 'ke', 'sawah,', 'tapi', 'beras', 'aja', 'impor', '@jokowi']	['mainnya', 'ke', 'sawah,', 'tapi', 'beras', 'aja', 'impor', 'jokowi']
2	['Ketum', 'PAN:', 'Stop', 'Impor', 'Beras,', 'Biarkan', 'Petani', 'Nikmati', 'Panen', '#ImporBeras']	['Ketum', 'PAN', 'Stop', 'Impor', 'Beras', 'Biarkan', 'Petani', 'Nikmati', 'Panen', 'ImporBeras']
3	['Impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat?', 'Bagus', 'tuh...', 'ntar', 'saat', 'pilpres', '2019', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru!', 'Jangan', 'Cuma', 'beras,', 'tapi', 'semua', 'komoditi', 'impor', '@jokowi']	['Impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat', 'Bagus', 'tuh', 'ntar', 'saat', 'pilpres', '2019', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru', 'Jangan', 'Cuma', 'beras', 'tapi', 'semua', 'komoditi', 'impor', 'jokowi']

4.7.3 Manualisasi Case Folding

Pada manualisasi *case folding* bertujuan untuk mengubah semua karakter yang terdapat huruf besar di setiap kata dalam suatu dokumen menjadi *lowercase* atau huruf kecil. pada Tabel 4.7 dan Tabel 4.8 menunjukkan manualisasi *case folding*.

Tabel 4.7 Manualisasi Case Folding (Data Latih)

NO	Data latih tweet	Hasil data latih tweet Case Folding
1	['Diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']	['diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']
2	['Masalah', 'Anda', 'ibarat', 'roda', 'Kadang', 'kempes', 'kadang', 'aman', 'Begitulah', 'hidupmu', 'Maklumlah', 'kamu', 'hanya', 'sebagai', 'Rakyat', 'Ikut', 'saja', 'aturan', 'mainnya.', 'selesai', 'saatnya', 'jalan', 'jalankan', 'saatnya', 'bayar', 'yoo', 'bayarlah', 'imporberas', 'pilihan2018']	['masalah', 'anda', 'ibarat', 'roda', 'kadang', 'kempes', 'kadang', 'aman', 'begitulah', 'hidupmu', 'maklumlah', 'kamu', 'hanya', 'sebagai', 'rakyat', 'ikut', 'saja', 'aturan', 'mainnya', 'selesai', 'saatnya', 'jalan', 'jalankan', 'saatnya', 'bayar', 'yoo', 'bayarlah', 'imporberas', 'pilihan']
3	['Negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'Sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas']	['negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'importerus', 'imporberas']
4	['Dianggap', 'Mengkhianati', 'Petani', 'MPR-DPR', 'Desak', 'Jokowi', 'Hentikan', 'Impor', 'Beras', 'ekonomi', 'imporberas']	['dianggap', 'mengkhianati', 'petani', 'mprdpr', 'desak', 'jokowi', 'hentikan', 'impor', 'beras', 'ekonomi', 'imporberas']
5	['BI', 'Dukung', 'Kran', 'Impor', 'Beras', 'ke', 'Batam', 'Dibuka', 'imporberas']	['bi', 'dukung', 'kran', 'impor', 'beras', 'ke', 'batam', 'dibuka', 'imporberas']
6	['Rizal', 'Ramli', 'Surya', 'Paloh', 'Dukung', 'Jokowi', 'Tetapi', 'Menterinya', 'Gerogotin', 'Elektabilitas', 'ImporBeras']	['rizal', 'ramli', 'surya', 'paloh', 'dukung', 'jokowi', 'tetapi', 'menterinya', 'gerogotin', 'elektabilitas', 'imporberas']

Tabel 4.8 Manualisasi Case Folding (Data Uji)

NO	Data uji tweet	Hasil data uji tweet Case Folding
1	['mainnya', 'ke', 'sawah', 'tapi', 'beras', 'aja', 'impor', 'jokowi']	['mainnya', 'ke', 'sawah', 'tapi', 'beras', 'aja', 'impor', 'jokowi']

2	['Ketum', 'PAN', 'Stop', 'Impor', 'Beras', 'Biarkan', 'Petani', 'Nikmati', 'Panen', 'ImporBeras']	['ketum', 'pan', 'stop', 'impor', 'beras', 'biarkan', 'petani', 'nikmati', 'panen', 'imporberas']
3	['Impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat', 'Bagus', 'tuh', 'ntar', 'saat', 'pilpres', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru', 'Jangan', 'Cuma', 'beras', 'tapi', 'semua', 'komoditi', 'impor', 'jokowi']	['Impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat', 'bagus', 'tuh', 'ntar', 'saat', 'pilpres', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru', 'jangan', 'cuma', 'beras', 'tapi', 'semua', 'komoditi', 'impor', 'jokowi']

4.7.4 Manualisasi *Filterisasi*

Pada manualisasi *filterisasi* bertujuan untuk menghapuskan kata yang tidak terdapat makna khusus atau tidak penting dalam suatu dokumen. Manualisasi *filterisasi* data latih dan data uji terdapat pada Tabel 4.9 dan Tabel 4.10.

Tabel 4.9 Manualisasi *Filterisasi* (Data Latih)

NO	Data latih <i>tweet</i>	Hasil data latih <i>Filterisasi</i>
1	['diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']	['diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']
2	['masalah', 'anda', 'ibarat', 'roda', 'Kadang', 'kempes', 'kadang', 'aman', 'begitulah', 'hidupmu', 'maklumlah', 'kamu', 'hanya', 'sebagai', 'rakyat', 'Ikut', 'saja', 'aturan', 'mainnya', 'selesai', 'saatnya', 'jalan', 'jalankan', 'saatnya', 'bayar', 'yoo', 'bayarlah', 'imporberas', 'pilihan']	['roda', 'kadang', 'kempes', 'kadang', 'aman', 'hidupmu', 'maklumlah', 'rakyat', 'aturan', 'mainnya', 'selesai', 'jalan', 'jalankan', 'bayar', 'yoo', 'bayarlah', 'imporberas']
3	['negeri', 'yang', 'maju', 'harus', 'rajin', 'impor', 'biar', 'kapitalis', 'semakin', 'makmur', 'dan', 'kemiskinan', 'semakin', 'menjamur', 'sehingga', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'Importerus', 'imporberas']	['negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'kemiskinan', 'menjamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'importerus', 'imporberas']
4	['dianggap', 'mengkianati', 'petani', 'mprdpr', 'desak', 'jokowi', 'hentikan']	['dianggap', 'mengkianati', 'petani', 'desak', 'jokowi', 'hentikan', 'impor', 'beras', 'ekonomi', 'imporberas']

	'impor', 'beras', 'ekonomi', 'imporberas']	
5	['bi', 'dukung', 'kran', 'impor', 'beras', 'ke', 'batam', 'dibuka', 'imporberas']	['bi', 'dukung', 'kran', 'impor', 'beras', 'batam', 'dibuka', 'imporberas']
6	['rizal', 'ramli', 'surya', 'paloh', 'dukung', 'jokowi', 'tetapi', 'menterinya', 'gerogotin', 'elektabilitas', 'imporberas']	['rizal', 'ramli', 'surya', 'paloh', 'dukung', 'jokowi', 'menterinya', 'gerogotin', 'elektabilitas', 'imporberas']

Tabel 4.10 Manualisasi *Filterisasi* (Data Uji)

NO	Data uji tweet	Hasil data uji tweet <i>Filterisasi</i>
1	['mainnya', 'ke', 'sawah', 'tapi', 'beras', 'aja', 'impor', 'jokowi']	['mainnya', 'sawah', 'beras', 'aja', 'impor']
2	['ketum', 'pan', 'stop', 'impor', 'beras', 'biarkan', 'petani', 'nikmati', 'panen', 'imporberas']	['ketum', 'pan', 'stop', 'impor', 'beras', 'biarkan', 'petani', 'nikmati', 'panen', 'imporberas']
3	['impor', 'beras', 'untuk', 'kesejahteraan', 'rakyat', 'bagus', 'tuh', 'ntar', 'saat', 'pilpres', 'saya', 'sarankan', 'itu', 'jadi', 'motto', 'baru', 'jangan', 'cuma', 'beras', 'tapi', 'semua', 'komoditi', 'impor', 'jokowi']	['impor', 'beras', 'kesejahteraan', 'rakyat', 'bagus', 'tuh', 'ntar', 'pilpres', 'sarankan', 'motto', 'beras', 'komoditi', 'impor']

4.7.5 Manualisasi *Stemming*

Pada manualisasi *stemming* merupakan tahap akhir dalam pemrosesan teks dari hasil proses *filterisasi*. Bertujuan untuk mengubah kata berimbuhan menjadi kata dasar dalam suatu dokumen. Tabel 4.11 dan Tabel 4.12 menunjukkan manualisasi *stemming*.

Tabel 4.11 Manualisasi *Stemming* (Data Latih)

NO	Data latih tweet	Hasil data latih tweet <i>Stemming</i>
1	['diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']	['diem', 'diem', 'aja', 'bro', 'imporberas', 'imporkeprihatinan']
2	['roda', 'kadang', 'kempes', 'kadang', 'aman', 'hidupmu', 'maklumlah', 'rakyat', 'aturan', 'mainnya', 'selesai', 'jalan', 'jalankan', 'bayar', 'yoo', 'bayarlah', 'imporberas']	['roda', 'kadang', 'kempes', 'kadang', 'aman', 'hidup', 'maklum', 'rakyat', 'atur', 'main', 'selesai', 'jalan', 'jalan', 'bayar', 'yoo', 'bayar', 'imporberas']

3	['negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'kemiskinan', 'menjamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'kelaparan', 'diatas', 'rerumputan', 'importerus', 'imporberas']	['negeri', 'maju', 'rajin', 'impor', 'biar', 'kapitalis', 'makmur', 'miskin', 'jamur', 'lambat', 'laun', 'bak', 'sapi', 'mati', 'lapar', 'atas', 'rumput', 'importerus', 'imporberas']
4	['dianggap', 'mengkhianati', 'petani', 'desak', 'jokowi', 'hentikan', 'impor', 'beras', 'ekonomi', 'imporberas']	['anggap', 'khianat', 'tani', 'desak', 'jokowi', 'henti', 'impor', 'beras', 'ekonomi', 'imporberas']
5	['bi', 'dukung', 'kran', 'impor', 'beras', 'batam', 'dibuka', 'imporberas']	['bi', 'dukung', 'kran', 'impor', 'beras', 'batam', 'buka', 'imporberas']
6	['rizal', 'ramli', 'surya', 'paloh', 'dukung', 'jokowi', 'menterinya', 'gerogotin', 'elektabilitas', 'imporberas']	['rizal', 'ramli', 'surya', 'paloh', 'dukung', 'jokowi', 'menteri', 'gerogotin', 'elektabilitas', 'imporberas']

Tabel 4.12 Manualisasi Stemming (Data Uji)

NO	Data uji tweet	Hasil data uji tweet Stemming
1	['mainnya', 'sawah', 'beras', 'aja', 'impor']	['main', 'sawah', 'beras', 'aja', 'impor']
2	['ketum', 'pan', 'stop', 'impor', 'beras', 'biarkan', 'petani', 'nikmati', 'panen', 'imporberas']	['tum', 'pan', 'stop', 'impor', 'beras', 'biar', 'tani', 'nikmat', 'panen', 'imporberas']
3	['impor', 'beras', 'kesejahteraan', 'rakyat', 'bagus', 'tuh', 'ntar', 'pilpres', 'sarankan', 'motto', 'beras', 'komoditi', 'impor']	['impor', 'beras', 'sejahtera', 'rakyat', 'bagus', 'tuh', 'ntar', 'pilpres', 'saran', 'motto', 'beras', 'komoditi', 'impor']

4.7.6 Manualisasi Pembobotan Kata TF, DF, W_{tf} dan IDF

Tahap manualisasi ini bertujuan untuk melakukan proses menghitung TF, DF, W_{tf} dan IDF di data latih dan data uji. Nilai TF merupakan *term* yang muncul di dalam dokumen, sedangkan nilai DF merupakan banyaknya dokumen yang mengandung kata tersebut. Tabel 4.13 menunjukkan data latih (D), dan data uji (T) untuk manualisasi TF dan DF.

Tabel 4.13 Manualisasi TF dan DF

Term	TF									df
	D1	D2	D3	D4	D5	D6	T1	T2	T3	
diem	2	0	0	0	0	0	0	0	0	1

aja	1	0	0	0	0	0	1	0	0	2
bro	1	0	0	0	0	0	0	0	0	1
imporberas	1	1	1	1	1	1	0	1	0	7
imporkeprihatinan	1	0	0	0	0	0	0	0	0	1
roda	0	1	0	0	0	0	0	0	0	1
kadang	0	2	0	0	0	0	0	0	0	1
kempes	0	1	0	0	0	0	0	0	0	1
aman	0	1	0	0	0	0	0	0	0	1
hidup	0	1	0	0	0	0	0	0	0	1
maklum	0	1	0	0	0	0	0	0	0	1
rakyat	0	1	0	0	0	0	0	0	1	1
atur	0	1	0	0	0	0	0	0	0	1
main	0	1	0	0	0	0	1	0	0	1
selesai	0	1	0	0	0	0	0	0	0	1
jalan	0	2	0	0	0	0	0	0	0	1
bayar	0	2	0	0	0	0	0	0	0	1
yoo	0	1	0	0	0	0	0	0	0	1
negeri	0	0	1	0	0	0	0	0	0	1
maju	0	0	1	0	0	0	0	0	0	1
rajin	0	0	1	0	0	0	0	0	0	1
impor	0	0	1	1	1	0	1	1	2	6
biar	0	0	1	0	0	0	0	1	0	2
kapitalis	0	0	1	0	0	0	0	0	0	1
makmur	0	0	1	0	0	0	0	0	0	1
misikin	0	0	1	0	0	0	0	0	0	1
jamur	0	0	1	0	0	0	0	0	0	1
lambat	0	0	1	0	0	0	0	0	0	1
laun	0	0	1	0	0	0	0	0	0	1
bak	0	0	1	0	0	0	0	0	0	1
sapi	0	0	1	0	0	0	0	0	0	1
mati	0	0	1	0	0	0	0	0	0	1
lapar	0	0	1	0	0	0	0	0	0	1
atas	0	0	1	0	0	0	0	0	0	1
rumput	0	0	1	0	0	0	0	0	0	1
importerus	0	0	1	0	0	0	0	0	0	1
anggap	0	0	0	1	0	0	0	0	0	1
khianat	0	0	0	1	0	0	0	0	0	1
tani	0	0	0	1	0	0	0	1	0	2
desak	0	0	0	1	0	0	0	0	0	1
jokowi	0	0	0	1	0	1	0	0	0	2
henti	0	0	0	1	0	0	0	0	0	1
beras	0	0	0	1	1	0	1	1	2	5

ekonomi	0	0	0	1	0	0	0	0	0	1
bi	0	0	0	0	1	0	0	0	0	1
dukung	0	0	0	0	1	1	0	0	0	2
kran	0	0	0	0	1	0	0	0	0	1
batam	0	0	0	0	1	0	0	0	0	1
buka	0	0	0	0	1	0	0	0	0	1
rizal	0	0	0	0	0	1	0	0	0	1
ramli	0	0	0	0	0	1	0	0	0	1
surya	0	0	0	0	0	1	0	0	0	1
paloh	0	0	0	0	0	1	0	0	0	1
menteri	0	0	0	0	0	1	0	0	0	1
gerogotin	0	0	0	0	0	1	0	0	0	1
elektabilitas	0	0	0	0	0	1	0	0	0	1
sawah	0	0	0	0	0	0	1	0	0	1
tum	0	0	0	0	0	0	0	1	0	1
pan	0	0	0	0	0	0	0	1	0	1
stop	0	0	0	0	0	0	0	1	0	1
nikmat	0	0	0	0	0	0	0	1	0	1
panen	0	0	0	0	0	0	0	1	0	1
sejahtera	0	0	0	0	0	0	0	0	1	1
bagus	0	0	0	0	0	0	0	0	1	1
tuh	0	0	0	0	0	0	0	0	1	1
ntar	0	0	0	0	0	0	0	0	1	1
pilpres	0	0	0	0	0	0	0	0	1	1
saran	0	0	0	0	0	0	0	0	1	1
motto	0	0	0	0	0	0	0	0	1	1
komoditi	0	0	0	0	0	0	0	0	1	1

Setelah melakukan perhitungan TF dan IDF. Untuk selanjutnya melakukan hitung nilai W_{tf} dan IDF untuk data latih dan data uji. W_{tf} merupakan bobot log frekuensi dari perolehan hasil nilai TF dan nilai W_{tf} dihitung menggunakan rumus pada Persamaan 2.1. IDF merupakan frekuensi munculnya kata pada seluruh dokumen. Nilai IDF dihitung menggunakan rumus pada Persamaan 2.2. Manualisasi W_{tf} dan IDF pada data latih dan data uji terdapat pada Tabel 4.14 menunjukkan manualisasi W_{tf} dan IDF.

Tabel 4.14 Manualisasi W_{tf} dan IDF

Term	TF									IDF
	D1	D2	D3	D4	D5	D6	T1	T2	T3	
diem	1,301	0	0	0	0	0	0	0	0	0,954243
aja	1	0	0	0	0	0	1	0	0	0,653213
bro	1	0	0	0	0	0	0	0	0	0,954243

imporberas	1	1	1	1	1	1	0	1	0	0,109144
imporkeprihatinan	1	0	0	0	0	0	0	0	0	0,954243
roda	0	1	0	0	0	0	0	0	0	0,954243
kadang	0	1,301	0	0	0	0	0	0	0	0,954243
kempes	0	1	0	0	0	0	0	0	0	0,954243
aman	0	1	0	0	0	0	0	0	0	0,954243
hidup	0	1	0	0	0	0	0	0	0	0,954243
maklum	0	1	0	0	0	0	0	0	0	0,954243
rakyat	0	1	0	0	0	0	0	0	1	0,653213
atur	0	1	0	0	0	0	0	0	0	0,954243
main	0	1	0	0	0	0	1	0	0	0,653213
selesai	0	1	0	0	0	0	0	0	0	0,954243
jalan	0	1,301	0	0	0	0	0	0	0	0,954243
bayar	0	1,301	0	0	0	0	0	0	0	0,954243
yoo	0	1	0	0	0	0	0	0	0	0,954243
negeri	0	0	1	0	0	0	0	0	0	0,954243
maju	0	0	1	0	0	0	0	0	0	0,954243
rajin	0	0	1	0	0	0	0	0	0	0,954243
impor	0	0	1	1	1	0	1	1	1,301	0,176091
biar	0	0	1	0	0	0	0	1	0	0,653213
kapitalis	0	0	1	0	0	0	0	0	0	0,954243
makmur	0	0	1	0	0	0	0	0	0	0,954243
miskin	0	0	1	0	0	0	0	0	0	0,954243
jamur	0	0	1	0	0	0	0	0	0	0,954243
lambat	0	0	1	0	0	0	0	0	0	0,954243
laun	0	0	1	0	0	0	0	0	0	0,954243
bak	0	0	1	0	0	0	0	0	0	0,954243
sapi	0	0	1	0	0	0	0	0	0	0,954243
mati	0	0	1	0	0	0	0	0	0	0,954243
lapar	0	0	1	0	0	0	0	0	0	0,954243
atas	0	0	1	0	0	0	0	0	0	0,954243
rumput	0	0	1	0	0	0	0	0	0	0,954243
importerus	0	0	1	0	0	0	0	0	0	0,954243
anggap	0	0	0	1	0	0	0	0	0	0,954243
khianat	0	0	0	1	0	0	0	0	0	0,954243
tani	0	0	0	1	0	0	0	1	0	0,653213
desak	0	0	0	1	0	0	0	0	0	0,954243
jokowi	0	0	0	1	0	1	0	0	0	0,653213
henti	0	0	0	1	0	0	0	0	0	0,954243
beras	0	0	0	1	1	0	1	1	1,301	0,255273
ekonomi	0	0	0	1	0	0	0	0	0	0,954243
bi	0	0	0	0	1	0	0	0	0	0,954243

dukung	0	0	0	0	1	1	0	0	0	0,653213
kran	0	0	0	0	1	0	0	0	0	0,954243
batam	0	0	0	0	1	0	0	0	0	0,954243
buka	0	0	0	0	1	0	0	0	0	0,954243
rizal	0	0	0	0	0	1	0	0	0	0,954243
ramli	0	0	0	0	0	1	0	0	0	0,954243
surya	0	0	0	0	0	1	0	0	0	0,954243
paloh	0	0	0	0	0	1	0	0	0	0,954243
menteri	0	0	0	0	0	1	0	0	0	0,954243
gerogotin	0	0	0	0	0	1	0	0	0	0,954243
elektabilitas	0	0	0	0	0	1	0	0	0	0,954243
sawah	0	0	0	0	0	0	1	0	0	0,954243
tum	0	0	0	0	0	0	0	1	0	0,954243
pan	0	0	0	0	0	0	0	1	0	0,954243
stop	0	0	0	0	0	0	0	1	0	0,954243
nikmat	0	0	0	0	0	0	0	1	0	0,954243
panen	0	0	0	0	0	0	0	1	0	0,954243
sejahtera	0	0	0	0	0	0	0	0	1	0,954243
bagus	0	0	0	0	0	0	0	0	1	0,954243
tuh	0	0	0	0	0	0	0	0	1	0,954243
ntar	0	0	0	0	0	0	0	0	1	0,954243
pilpres	0	0	0	0	0	0	0	0	1	0,954243
saran	0	0	0	0	0	0	0	0	1	0,954243
motto	0	0	0	0	0	0	0	0	1	0,954243
komoditi	0	0	0	0	0	0	0	0	1	0,954243

4.7.7 Manualisasi TF-IDF

Manualisasi ini merupakan tahap akhir dalam melakukan perhitungan pembobotan kata yang bertujuan untuk menghitung nilai TF-IDF. Tahapan ini diproses setelah melakukan proses perhitungan TF, DF, W_{tf} dan IDF. Perhitungan nilai TF-IDF menggunakan Persamaan 2.4. Tabel 4.15 Manualisasi TF-IDF pada data latih dan data uji.

Tabel 4.15 Manualisasi TF-IDF

Term	TF.IDF									IDF
	D1	D2	D3	D4	D5	D6	T1	T2	T3	
diem	1,241	0	0	0	0	0	0	0	0	0,954243
aja	0,653	0	0	0	0	0	0,653	0	0	0,653213
bro	0,954	0	0	0	0	0	0	0	0	0,954243
imporberas	0,109	0,1091	0,109	0,109	0,1091	0,1091	0	0,109	0	0,109144
imporkeprihatinan	0,954	0	0	0	0	0	0	0	0	0,954243

roda	0	0,9542	0	0	0	0	0	0	0	0,954243
kadang	0	1,2415	0	0	0	0	0	0	0	0,954243
kempes	0	0,9542	0	0	0	0	0	0	0	0,954243
aman	0	0,9542	0	0	0	0	0	0	0	0,954243
hidup	0	0,9542	0	0	0	0	0	0	0	0,954243
maklum	0	0,9542	0	0	0	0	0	0	0	0,954243
rakyat	0	0,6532	0	0	0	0	0	0	0,653	0,653213
atur	0	0,9542	0	0	0	0	0	0	0	0,954243
main	0	0,6532	0	0	0	0	0,653	0	0	0,653213
selesai	0	0,9542	0	0	0	0	0	0	0	0,954243
jalan	0	1,2415	0	0	0	0	0	0	0	0,954243
bayar	0	1,2415	0	0	0	0	0	0	0	0,954243
yoo	0	0,9542	0	0	0	0	0	0	0	0,954243
negeri	0	0	0,954	0	0	0	0	0	0	0,954243
maju	0	0	0,954	0	0	0	0	0	0	0,954243
rajin	0	0	0,954	0	0	0	0	0	0	0,954243
impor	0	0	0,176	0,176	0,1761	0	0,176	0,176	0,229	0,176091
biar	0	0	0,653	0	0	0	0	0,653	0	0,653213
kapitalis	0	0	0,954	0	0	0	0	0	0	0,954243
makmur	0	0	0,954	0	0	0	0	0	0	0,954243
miskin	0	0	0,954	0	0	0	0	0	0	0,954243
jamur	0	0	0,954	0	0	0	0	0	0	0,954243
lambat	0	0	0,954	0	0	0	0	0	0	0,954243
laun	0	0	0,954	0	0	0	0	0	0	0,954243
bak	0	0	0,954	0	0	0	0	0	0	0,954243
sapi	0	0	0,954	0	0	0	0	0	0	0,954243
mati	0	0	0,954	0	0	0	0	0	0	0,954243
lapar	0	0	0,954	0	0	0	0	0	0	0,954243
atas	0	0	0,954	0	0	0	0	0	0	0,954243
rumput	0	0	0,954	0	0	0	0	0	0	0,954243
importerus	0	0	0,954	0	0	0	0	0	0	0,954243
anggap	0	0	0	0,954	0	0	0	0	0	0,954243
khianat	0	0	0	0,954	0	0	0	0	0	0,954243
tani	0	0	0	0,653	0	0	0	0,653	0	0,653213
desak	0	0	0	0,954	0	0	0	0	0	0,954243
jokowi	0	0	0	0,653	0	0,6532	0	0	0	0,653213
henti	0	0	0	0,954	0	0	0	0	0	0,954243
beras	0	0	0	0,255	0,2553	0	0,255	0,255	0,332	0,255273
ekonomi	0	0	0	0,954	0	0	0	0	0	0,954243
bi	0	0	0	0	0,9542	0	0	0	0	0,954243
dukung	0	0	0	0	0,6532	0,6532	0	0	0	0,653213
kran	0	0	0	0	0,9542	0	0	0	0	0,954243
batam	0	0	0	0	0,9542	0	0	0	0	0,954243

buka	0	0	0	0	0,9542	0	0	0	0	0,954243
rizal	0	0	0	0	0	0,9542	0	0	0	0,954243
ramli	0	0	0	0	0	0,9542	0	0	0	0,954243
surya	0	0	0	0	0	0,9542	0	0	0	0,954243
paloh	0	0	0	0	0	0,9542	0	0	0	0,954243
menteri	0	0	0	0	0	0,9542	0	0	0	0,954243
gerogotin	0	0	0	0	0	0,9542	0	0	0	0,954243
elektabilitas	0	0	0	0	0	0,9542	0	0	0	0,954243
sawah	0	0	0	0	0	0	0,954	0	0	0,954243
tum	0	0	0	0	0	0	0	0,954	0	0,954243
pan	0	0	0	0	0	0	0	0,954	0	0,954243
stop	0	0	0	0	0	0	0	0,954	0	0,954243
nikmat	0	0	0	0	0	0	0	0,954	0	0,954243
panen	0	0	0	0	0	0	0	0,954	0	0,954243
sejahtera	0	0	0	0	0	0	0	0	0,954	0,954243
bagus	0	0	0	0	0	0	0	0	0,954	0,954243
tuh	0	0	0	0	0	0	0	0	0,954	0,954243
ntar	0	0	0	0	0	0	0	0	0,954	0,954243
pilpres	0	0	0	0	0	0	0	0	0,954	0,954243
saran	0	0	0	0	0	0	0	0	0,954	0,954243
motto	0	0	0	0	0	0	0	0	0,954	0,954243
komoditi	0	0	0	0	0	0	0	0	0,954	0,954243

4.7.8 Manualisasi Support Vector Machine

Tahap manualisasi ini merupakan metode klasifikasi yang dilakukan ketika selesai melakukan perhitungan proses pembobotan kata nilai TF-IDF. Tahap awal dalam memproses klasifikasi SVM yaitu mencari *hyperplane* atau fungsi *sign (h(x))* melalui *Sequential Training SVM*. Berikut tahap yang dilakukan pada manualisasi *Support Vector Machine*

1. Menginisialisasikan beberapa variabel untuk diproses. Variabel variabel tersebut antara lain:

$c = 1$ (konstanta untuk perhitungan kernel)

$\lambda = 0,5$

$\gamma = 0,001$ (*learning rate*)

$C = 1$ (variable *slack*)

$\varepsilon = 0,0001$ (epsilon)

Iterasi maksimum = 10

2. Berikutnya inialisasi $\alpha_i = 0$ dan melakukan perhitungan matriks Hessian D_{ij} dengan Persamaan 2.6. Persamaan ini menggunakan kernel *polynomial*

dengan d atau *degree* = 2 yaitu: $K(x_i, x_j) = (x_i \cdot x_j + c)^d$. Perhitungan manual kernel *polynomial* pada x_1 (menandakan dokumen data ke-1) sebagai berikut

[illegible]

$$K(x_1, x_1) = (x_1 \cdot x_1 + 1)^2$$

$$K(x_1, x_1) =$$

[illegible]

$$K(x_1, x_1) = (3,80 + 1)^2$$

$$K(x_1, x_1) = 23,05$$

Hasil perhitungan kernel untuk seluruh dokumen ditunjukkan pada Tabel 4.15.

Tabel 4.16 Perhitungan Kernel Seluruh dokumen (data latih)

	D1	D2	D3	D4	D5	D6
D1	23,05	1,02	1,02	1,02	1,02	1,02
D2	1,02	189,7	1,02	1,02	1,02	1,02
D3	1,02	1,02	257,2	1,09	1,09	1,02
D4	1,02	1,02	1,09	42,4	1,23	2,07
D5	1,02	1,02	1,09	1,23	26,8	2,07
D6	1,02	1,02	1,02	2,07	2,07	67,89

Tahap berikutnya yaitu melakukan perhitungan Matriks Hessian pada dokumen data ke-1 atau x_1 dijabarkan sebagai berikut.

$$D_{ij} = y_i, y_j (K(x_i, x_j) + \lambda^2)$$

$$D_{11} = y_1, y_1 (K(x_1, x_1) + \lambda^2)$$

$$D_{11} = -1. -1 (23,05) + 0,5^2)$$

$$D_{11} = (23,05 + 0,25)$$

$$D_{11} = 23,30$$

Tabel 4.17 menunjukkan hasil perhitungan Matriks Hessian untuk seluruh dokumen.

Tabel 4.17 Perhitungan Matriks Hessian Seluruh dokumen (data latih)

	D1	D2	D3	D4	D5	D6
D1	23,30	-1,27	-1,27	1,27	-1,27	-1,27
D2	-1,27	189,969	1,27	-1,27	1,27	1,27
D3	-1,27	1,27	257,50	-1,34	1,34	1,27
D4	1,27	-1,27	-1,34	42,69	-1,48	-2,32
D5	-1,27	1,27	1,34	-1,48	27,05	2,32
D6	-1,27	1,27	1,27	-2,32	2,32	68,14

3. Tahap ini melakukan proses perhitungan pada nilai E_i , $\delta\alpha_i$, dan α_i sebagai berikut.

- a. Menghitung nilai E_i yang terdapat di Persamaan 2.7 dan hasil perhitungan Matriks Hessian. Perhitungan manual E_i dijabarkan sebagai berikut.

$$E_i = \sum_{j=0}^N \alpha_j D_{ij}$$

$$E_1 = (23,30 \cdot 0) + ((-1,27) \cdot 0) + ((-1,27) \cdot 0) + (1,27 \cdot 0) + ((-1,27) \cdot 0) + ((-0,27) \cdot 0)$$

$$E_1 = 0$$

Tabel 4.18. menunjukkan hasil nilai perhitungan E_i untuk seluruh dokumen data latih

Tabel 4.18 Hasil perhitungan E_i pada data latih

E_i	D1	D2	D3	D4	D5	D6
	0	0	0	0	0	0

- b. Setelah memperoleh nilai E_i , maka hitung nilai $\delta\alpha_i$ dengan Persamaan 2.8. perhitungan manual $\delta\alpha_i$ dijabarkan sebagai berikut.

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i); -\alpha_i]; C - \alpha_i\}$$

$$\delta\alpha_1 = \min\{\max[\gamma(1 - E_1); -\alpha_1]; C - \alpha_1\}$$

$$\delta\alpha_1 = \min\{\max[0,001(1 - 0); -0]; 1 - 0\}$$

$$\delta\alpha_1 = \min\{\max[0,001; 0]; 1\}$$

$$\delta\alpha_1 = \min\{\max 0,0001; 1\}$$

$$\delta\alpha_1 = 0,001$$

Tabel 4.19 menunjukkan hasil perhitungan $\delta\alpha_i$ pada seluruh dokumen data latih

Tabel 4.19 Hasil perhitungan $\delta\alpha$ pada data latih

$\delta\alpha_i$	D1	D2	D3	D4	D5	D6
	0,001	0,001	0,001	0,001	0,001	0,001

- c. Setelah memperoleh nilai $\delta\alpha_i$ tahap selanjutnya menghitung nilai α_i dengan Persamaan 2.9 Perhitungan α_i dijabarkan sebagai berikut.

$$\alpha_i = \alpha_i + \delta\alpha_i$$

$$\alpha_i = \alpha_i + \delta\alpha_1$$

$$\alpha_1 = 0 + 0,001$$

Hasil nilai perhitungan α_i pada seluruh dokumen data latih ditunjukkan pada Tabel 4.20.

Tabel 4.20 Hasil perhitungan antara α pada data latih

	D1	D2	D3	D4	D5	D6
α lama	0	0	0	0	0	0
α baru	0,001	0,001	0,001	0,001	0,001	0,001

4. Melakukan tahap ketiga hingga memperoleh jumlah iterasi maksimum atau $\text{Max}(|\delta\alpha|) < \varepsilon$. Batas atas iterasi adalah 10. Hasil perhitungan manualisasi dari E_i , $\delta\alpha_i$, dan α_i pada iterasi ke-10 ditunjukkan pada Tabel 4.21, 4.22 dan 4.23.

Tabel 4.21 Hasil perhitungan E saat memperoleh iterasi maksimum pada data latih

E_i	D1	D2	D3	D4	D5	D6
	0,16225	0,85196	0,932456	0,29140	0,23433	0,47661

Tabel 4.22 Hasil perhitungan $\delta\alpha$ saat mencapai iterasi maksimum pada data latih

$\delta\alpha_i$	D1	D2	D3	D4	D5	D6
	0,00083	0,000148	0,000068	0,00070	0,00076	0,00052

Tabel 4.23 Hasil perhitungan α saat mencapai iterasi maksimum pada data latih

	D1	D2	D3	D4	D5	D6
α lama	0,008332	0,004454	0,003603	0,007760	0,008016	0,006866
α baru	0,009167	0,004603	0,003670	0,008468	0,008782	0,007389

5. Setelah melakukan langkah diatas diketahui bahwa nilai α akhir dan *support vector* (SV), merupakan data yang mempunyai nilai $\alpha > 0$. Tabel 4.24. Hasil support vector yang didapatkan sebagai berikut

Tabel 4.24 Hasil *support vector*

SV	D1	D2	D3	D4	D5	D6
	0,009167	0,004602	0,003670	0,008468	0,008782	0,007389

6. Selanjutnya, menghitung nilai *bias* b yang didapatkan pada Persamaan 2.10. Untuk mengetahui x^+ dan x^- , menunjukkan dokumen yang mempunyai nilai α terbesar di kelas sentimennya. Perhitungan manual ini, x^+ tedapat di dokumen D5 dan x^- terdapat di dokumen D1. Perhitungan *bias* ditunjukkan pada Tabel 4.25.

Tabel 4.25 Hasil perhitungan nilai b (bias)

x_i	$(K(x_i, x^+))$	$(K(x_i, x^-))$	$\alpha_i y_i (K(x_i, x^+))$	$\alpha_i y_i (K(x_i, x^-))$
D1	1,024	23,050	-0,009387	-0,211313
D2	1,024	1,024	0,004713	0,004713
D3	1,088	1,024	0,003993	0,003759
D4	1,228	1,024	-0,010398	-0,008672
D5	26,802	1,024	0,235378	0,008993
D6	2,070	1,024	0,015294	0,007567
			$\sum_{i=0}^N \alpha_i y_i (K(x_i, x^+))$	$\sum_{i=0}^N \alpha_i y_i (K(x_i, x^-))$
			0,239593	-0,194953

$$b = -\frac{1}{2} (\sum_{i=0}^N \alpha_i y_i (K(x_i, x^-)) + \sum_{i=0}^N \alpha_i y_i (K(x_i, x^+)))$$

$$b = -\frac{1}{2} (0,239593 + (-0,194953))$$

$$b = -0,022319831$$

7. Tahap terakhir yang dilakukan yaitu *testing* atau memasukkan data uji ke *hyperplane* untuk memperoleh informasi hasil klasifikasi kelas sentimen data uji dengan menggunakan rumus pada Persamaan 2.11. Apabila memperoleh hasil $sign(h(x))$ bernilai -1, maka dokumen teks termasuk ke dalam kelas sentimen negatif. Namun jika hasil yang didapatkan memperoleh nilai $sign(h(x)) + 1$, maka dokumen teks termasuk ke dalam kelas sentimen positif. Tabel 4.26 menunjukkan perhitungan manualisasi dari *testing* atau klasifikasi SVM untuk data uji.

Tabel 4.26 Hasil perhitungan *testing Support Vector Machine* pada data uji

$\alpha_i y_i (K(x, x_i))$	T1	T2	T3
D1	-0,018659815	-0,009387201	-0,009167484
D2	0,009368928	0,004713231	0,009368928
D3	0,003902163	0,003992857	0,003973141
D4	-0,010176001	-0,019948348	-0,008982962
D5	0,010552453	0,010783055	0,011117210
D6	0,007389960	0,007567074	0,007389960
$\sum_{i=0}^N \alpha_i y_i (K(x, x_i))$	0,002377688	-0,002279332	0,013698792
$sign(h(x)) = \sum_{i=0}^N \alpha_i y_i (K(x, x_i)) + b$	-0,019942	-0,024599	-0,008621
Kelas Sentimen	Negatif	Negatif	negatif

8. Dalam tahap ini sebelum melakukan proses perhitungan penggabungan yaitu, lakukan perhitungan normalisasi jumlah *retweet* untuk memberikan bobot pada jumlah *retweet*, dengan cara jumlah *retweet* yang didapatkan pada dokumen *tweet*, akan melakukan normalisasi menggunakan min-max agar bobot yang diberikan seimbang. Normalisasi dilakukan untuk tiap data uji yang dibandingkan dengan data latih. Manualisasi pembobotan jumlah *retweet* terdapat pada Tabel 4.27.

Tabel 4.27 Manualisasi Pembobotan Jumlah *Retweet*

Dokumen Uji	Jumlah <i>Retweet</i>	Dokumen Latih	Jumlah <i>Retweet</i>	Normalisasi
T1	10	D1	0	0,9
		D2	0	
		D3	0	
		D4	0	
		D5	2	
		D6	4	

T2	3	D1	0	0,7
		D2	0	
		D3	0	
		D4	0	
		D5	2	
		D6	4	
T3	0	D1	0	0,1
		D2	0	
		D3	0	
		D4	0	
		D5	2	
		D6	4	

4.7.9 Manualisasi Penggabungan

Proses perhitungan manualisasi penggabungan ini merupakan tahap akhir untuk menghitung nilai dari pembobotan tekstual (hasil nilai klasifikasi metode SVM) dan bobot non-tekstual (jumlah *retweet*). Bobot tersebut akan melakukan perkalian pada konstanta yang diinisialisasikan α dan β . Manualisasi penggabungan terdapat pada manualisasi penggabungan terdapat pada Tabel 4.28.

Tabel 4.28 Manualisasi Penggabungan

α	β	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan	Hasil Penggabungan
	$(1-\alpha)$					Pos = ($\alpha * (\text{ntekstual})$) + ($\beta * \text{nnon-tekstual}$) Neg = ($\alpha * (\text{ntekstual})$)	
100%	0%	T1	Negatif	-0,020	0,9	Neg = ($1 * (-0,020)$) = -0,020	Neg = -0,020
			Positif			Pos = ($1 * (-0,020) + (0 * 0,9)$) = -0,020	Pos = -0,020
			Positif			Pos = ($1 * (-0,020) + (0 * 0,9)$) = -0,020	Pos = -0,020
			Negatif			Neg = ($1 * (-0,020)$) = -0,020	Neg = -0,020
			Positif			Pos = ($1 * (-0,020) + (0 * 0,9)$) = -0,020	Pos = -0,020
			Positif			Pos = ($1 * (-0,020) + (0 * 0,9)$) = -0,020	Pos = -0,020
		T2	Negatif	-0,021	0,7	Neg = ($1 * (-0,021)$) = -0,021	Neg = -0,021
			Positif			Pos = ($1 * (-0,021) + (0 * 0,7)$) = -0,021	Pos = -0,021
			Positif			Pos = ($1 * (-0,021) + (0 * 0,7)$) = -0,021	Pos = -0,021
			Negatif			Neg = ($1 * (-0,021)$) = -0,021	Neg = -0,021
			Positif			Pos = ($1 * (-0,021) + (0 * 0,7)$) = -0,021	Pos = -0,021
			Positif			Pos = ($1 * (-0,021) + (0 * 0,7)$) = -0,021	Pos = -0,021

α	β	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan	Hasil Penggabungan
	$(1-\alpha)$					Pos $= (\alpha * (\text{ntekstual})) + (B * (\text{nnon-tekstual}))$ Neg $= (\alpha * (\text{ntekstual}))$	
90%	10%	T3	Negatif	-0,010	0,1	Neg $= (1 * (-0,010)) = -0,010$	Neg = -0,010
			Positif			Pos $= (1 * (-0,010) + (0 * 0,1)) = -0,010$	Pos = -0,010
			Positif			Pos $= (1 * (-0,010) + (0 * 0,1)) = -0,010$	Pos = -0,010
			Negatif			Neg $= (1 * (-0,010)) = -0,010$	Neg = -0,010
			Positif			Pos $= (1 * (-0,010) + (0 * 0,1)) = -0,010$	Pos = -0,010
			Positif			Pos $= (1 * (-0,010) + (0 * 0,1)) = -0,010$	Pos = -0,010
		T1	Negatif	-0,020	0,9	Neg $= (0,9 * (-0,020)) = -0,018$	Neg = -0,018
			Positif			Pos $= (0,9 * (-0,020) + (0,1 * 0,9)) = -0,072$	Pos = -0,072
			Positif			Pos $= (0,9 * (-0,020) + (0,1 * 0,9)) = -0,072$	Pos = -0,072
			Negatif			Neg $= (0,9 * (-0,020)) = -0,018$	Neg = -0,018
			Positif			Pos $= (0,9 * (-0,020) + (0,1 * 0,9)) = -0,072$	Pos = -0,072
			Positif			Pos $= (0,9 * (-0,020) + (0,1 * 0,9)) = -0,072$	Pos = -0,072
		T2	Negatif	-0,021	0,7	Neg $= (0,9 * (-0,021)) = -0,019$	Neg = -0,019
			Positif			Pos $= (0,9 * (-0,021) + (0,1 * 0,9)) = -0,051$	Pos = -0,051
			Positif			Pos $= (0,9 * (-0,021) + (0,1 * 0,9)) = -0,051$	Pos = -0,051
			Negatif			Neg $= (0,9 * (-0,021)) = -0,019$	Neg = -0,019
			Positif			Pos $= (0,9 * (-0,021) + (0,1 * 0,9)) = -0,051$	Pos = -0,051
			Positif			Pos $= (0,9 * (-0,021) + (0,1 * 0,9)) = -0,051$	Pos = -0,051
		T3	Negatif	-0,010	0,1	Neg $= (0,9 * (-0,010)) = -0,009$	Neg = -0,009
			Positif			Pos $= (0,9 * (-0,010) + (0,1 * 0,1)) = 0,001$	Pos = 0,001
			Positif			Pos $= (0,9 * (-0,010) + (0,1 * 0,1)) = 0,001$	Pos = 0,001
			Negatif			Neg $= (0,9 * (-0,010)) = -0,009$	Neg = -0,009
			Positif			Pos $= (0,9 * (-0,010) + (0,1 * 0,1)) = 0,001$	Pos = 0,001
			Positif			Pos $= (0,9 * (-0,010) + (0,1 * 0,1)) = 0,001$	Pos = 0,001

α	β (1- α)	Dok. Uji	Kategori	Nilai SVM	Bobot <i>Retweet</i>	Perhitungan Penggabungan Pos = (α * (ntekstual)) + (β * (nnon-tekstual)) Neg = (α * (ntekstual))	Hasil Penggabungan
80%	20%	T1	Negatif	-0,020	0,9	Neg = (0,8 * (-0,020)) = -0,016	Neg = -0,016
			Positif			Pos = (0,8 * (-0,020)) + (0,2 * 0,9) = 0,164	Pos = 0,164
			Positif			Pos = (0,8 * (-0,020)) + (0,2 * 0,9) = 0,164	Pos = 0,164
			Negatif			Neg = (0,8 * (-0,020)) = -0,016	Neg = -0,016
			Positif			Pos = (0,8 * (-0,020)) + (0,2 * 0,9) = 0,164	Pos = 0,164
			Positif			Pos = (0,8 * (-0,020)) + (0,2 * 0,9) = 0,164	Pos = 0,164
		T2	Negatif	-0,021	0,7	Neg = (0,8 * (-0,021)) = -0,017	Neg = -0,017
			Positif			Pos = (0,8 * (-0,021)) + (0,2 * 0,9) = 0,123	Pos = 0,123
			Positif			Pos = (0,8 * (-0,021)) + (0,2 * 0,9) = 0,123	Pos = 0,123
			Negatif			Neg = (0,8 * (-0,021)) = -0,017	Neg = -0,017
			Positif			Pos = (0,8 * (-0,021)) + (0,2 * 0,9) = 0,123	Pos = 0,123
			Positif			Pos = (0,8 * (-0,021)) + (0,2 * 0,9) = 0,123	Pos = 0,123
		T3	Negatif	-0,010	0,1	Neg = (0,8 * (-0,010)) = -0,008	Neg = -0,008
			Positif			Pos = (0,8 * (-0,010)) + (0,2 * 0,1) = 0,012	Pos = 0,012
			Positif			Pos = (0,8 * (-0,010)) + (0,2 * 0,1) = 0,012	Pos = 0,012
			Negatif			Neg = (0,8 * (-0,010)) = -0,008	Neg = -0,008
			Positif			Pos = (0,8 * (-0,010)) + (0,2 * 0,1) = 0,012	Pos = 0,012
			Positif			Pos = (0,8 * (-0,010)) + (0,2 * 0,1) = 0,012	Pos = 0,012
70%	30%	T1	Negatif	-0,020	0,9	Neg = (0,7 * (-0,020)) = -0,014	Neg = -0,014
			Positif			Pos = (0,7 * (-0,020)) + (0,3 * 0,9) = 0,256	Pos = 0,256
			Positif			Pos = (0,7 * (-0,020)) + (0,3 * 0,9) = 0,256	Pos = 0,256
			Negatif			Neg = (0,7 * (-0,020)) = -0,014	Neg = -0,014
			Positif			Pos = (0,7 * (-0,020)) + (0,3 * 0,9) = 0,256	Pos = 0,256
			Positif			Pos = (0,7 * (-0,020)) + (0,3 * 0,9) = 0,256	Pos = 0,256

α	β (1- α)	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan Pos = (α * (ntekstual)) + (β * (nnon-tekstual)) Neg = (α * (ntekstual))	Hasil Penggabungan
			Positif	-0,020	0,9	Pos = (0,7*(-0,020)+(0,3*0,9) = 0,256	Pos = 0,256
70%	30%	T2	Negatif	-0,021	0,7	Neg = (0,7*(-0,021) = -0,014	Neg = -0,014
			Positif			Pos = (0,7*(-0,021)+(0,3*0,9) = 0,196	Pos = 0,196
			Positif			Pos = (0,7*(-0,021)+(0,3*0,9) = 0,196	Pos = 0,196
			Negatif			Neg = (0,7*(-0,021) = -0,014	Neg = -0,014
			Positif			Pos = (0,7*(-0,021)+(0,3*0,9) = 0,196	Pos = 0,196
			Positif			Pos = (0,7*(-0,021)+(0,3*0,9) = 0,196	Pos = 0,196
		T3	Negatif	-0,010	0,1	Neg = (0,7*(-0,010) = -0,007	Neg = -0,007
			Positif			Pos = (0,7*(-0,010)+(0,3*0,1) = 0,023	Pos = 0,023
			Positif			Pos = (0,7*(-0,010)+(0,3*0,1) = 0,023	Pos = 0,023
			Negatif			Neg = (0,7*(-0,010) = -0,007	Neg = -0,007
			Positif			Pos = (0,7*(-0,010)+(0,3*0,1) = 0,023	Pos = 0,023
			Positif			Pos = (0,7*(-0,010)+(0,3*0,1) = 0,023	Pos = 0,023
60%	40%	T1	Negatif	-0,020	0,9	Neg = (0,6*(-0,020) = -0,012	Neg = -0,012
			Positif			Pos = (0,6*(-0,020)+(0,4*0,9) = 0,348	Pos = 0,348
			Positif			Pos = (0,6*(-0,020)+(0,4*0,9) = 0,348	Pos = 0,348
			Negatif			Neg = (0,6*(-0,020) = -0,012	Neg = -0,012
			Positif			Pos = (0,6*(-0,020)+(0,4*0,9) = 0,348	Pos = 0,348
			Positif			Pos = (0,6*(-0,020)+(0,4*0,9) = 0,348	Pos = 0,348
		T2	Negatif	-0,021	0,7	Neg = (0,6*(-0,021) = -0,012	Neg = -0,012
			Positif			Pos = (0,6*(-0,021)+(0,4*0,9) = 0,268	Pos = 0,268
			Positif			Pos = (0,6*(-0,021)+(0,4*0,9) = 0,268	Pos = 0,268
			Negatif			Neg = (0,6*(-0,021) = -0,012	Neg = -0,012

α	β (1- α)	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan Pos = (α * (ntekstual)) + (β * (nnon- tekstual)) Neg = (α * (ntekstual))	Hasil Penggabungan
60%	40%	T2	Positif	- 0,021	0,7	Pos = (0,6 * (-0,021) + (0,4 * 0,9) = 0,268	Pos = 0,268
			Positif			Pos = (0,6 * (-0,021) + (0,4 * 0,9) = 0,268	Pos = 0,268
		T3	Negatif	- 0,010	0,1	Neg = (0,6 * (-0,010) = -0,006	Neg = -0,006
			Positif			Pos = (0,6 * (-0,010) + (0,4 * 0,1) = 0,034	Pos = 0,034
			Positif			Pos = (0,6 * (-0,010) + (0,4 * 0,1) = 0,034	Pos = 0,034
			Negatif			Neg = (0,6 * (-0,010) = -0,006	Neg = -0,006
			Positif			Pos = (0,6 * (-0,010) + (0,4 * 0,1) = 0,034	Pos = 0,034
			Positif			Pos = (0,6 * (-0,010) + (0,4 * 0,1) = 0,034	Pos = 0,034
50%	50%	T1	Negatif	- 0,020	0,9	Neg = (0,5 * (-0,020) = -0,010	Neg = -0,010
			Positif			Pos = (0,5 * (-0,020) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Positif			Pos = (0,5 * (-0,020) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Negatif			Neg = (0,5 * (-0,020) = -0,010	Neg = -0,010
			Positif			Pos = (0,5 * (-0,020) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Positif			Pos = (0,5 * (-0,020) + (0,5 * 0,9) = 0,440	Pos = 0,440
		T2	Negatif	- 0,021	0,7	Neg = (0,5 * (-0,021) = -0,010	Neg = -0,010
			Positif			Pos = (0,5 * (-0,021) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Positif			Pos = (0,5 * (-0,021) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Negatif			Neg = (0,5 * (-0,021) = -0,010	Neg = -0,010
			Positif			Pos = (0,5 * (-0,021) + (0,5 * 0,9) = 0,440	Pos = 0,440
			Positif			Pos = (0,5 * (-0,021) + (0,5 * 0,9) = 0,440	Pos = 0,440
		T3	Negatif	- 0,010	0,1	Neg = (0,5 * (-0,010) = -0,005	Neg = -0,005
			Positif			Pos = (0,5 * (-0,010) + (0,5 * 0,1) = 0,045	Pos = 0,045

α	β (1- α)	Dok. Uji	Kategori	Nilai SVM	Bobot <i>Retweet</i>	Perhitungan Penggabungan Pos = (α * (ntekstual)) + (β * (nnon-tekstual)) Neg = (α * (ntekstual))	Hasil Penggabungan
50%	50%	T3	Positif	-0,010	0,1	Pos = (0,5 * (-0,010)) + (0,5 * 0,1) = 0,045	Pos = 0,045
			Negatif			Neg = (0,5 * (-0,010)) = -0,005	Neg = -0,005
			Positif			Pos = (0,5 * (-0,010)) + (0,5 * 0,1) = 0,045	Pos = 0,045
			Positif			Pos = (0,5 * (-0,010)) + (0,5 * 0,1) = 0,045	Pos = 0,045
40%	60%	T1	Negatif	-0,020	0,9	Neg = (0,4 * (-0,020)) = -0,008	Neg = -0,008
			Positif			Pos = (0,4 * (-0,020)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Positif			Pos = (0,4 * (-0,020)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Negatif			Neg = (0,4 * (-0,020)) = -0,008	Neg = -0,008
			Positif			Pos = (0,4 * (-0,020)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Positif			Pos = (0,4 * (-0,020)) + (0,6 * 0,9) = 0,532	Pos = 0,532
		T2	Negatif	-0,021	0,7	Neg = (0,4 * (-0,021)) = -0,008	Neg = -0,008
			Positif			Pos = (0,4 * (-0,021)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Positif			Pos = (0,4 * (-0,021)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Negatif			Neg = (0,4 * (-0,021)) = -0,008	Neg = -0,008
			Positif			Pos = (0,4 * (-0,021)) + (0,6 * 0,9) = 0,532	Pos = 0,532
			Positif			Pos = (0,4 * (-0,021)) + (0,6 * 0,9) = 0,532	Pos = 0,532
		T3	Negatif	-0,010	0,1	Neg = (0,4 * (-0,010)) = -0,004	Neg = -0,004
			Positif			Pos = (0,4 * (-0,010)) + (0,6 * 0,1) = 0,056	Pos = 0,056
			Positif			Pos = (0,4 * (-0,010)) + (0,6 * 0,1) = 0,056	Pos = 0,056
			Negatif			Neg = (0,4 * (-0,010)) = -0,004	Neg = -0,004
			Positif			Pos = (0,4 * (-0,010)) + (0,6 * 0,1) = 0,056	Pos = 0,056
			Positif			Pos = (0,4 * (-0,010)) + (0,6 * 0,1) = 0,056	Pos = 0,056

α	β (1- α)	Dok. Uji	Kategori	Nilai SVM	Bobot <i>Retweet</i>	Perhitungan Penggabungan Pos = (α * (ntekstual)) + (β * (nnon-tekstual)) Neg = (α * (ntekstual))	Hasil Penggabungan
30%	70%	T1	Negatif	-0,020	0,9	Neg = ($0,3 * (-0,020)$) = -0,006	Neg = -0,006
			Positif			Pos = ($0,3 * (-0,020) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Positif			Pos = ($0,3 * (-0,020) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Negatif			Neg = ($0,3 * (-0,020)$) = -0,006	Neg = -0,006
			Positif			Pos = ($0,3 * (-0,020) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Positif			Pos = ($0,3 * (-0,020) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
		T2	Negatif	-0,021	0,7	Neg = ($0,3 * (-0,021)$) = -0,006	Neg = -0,006
			Positif			Pos = ($0,3 * (-0,021) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Positif			Pos = ($0,3 * (-0,021) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Negatif			Neg = ($0,3 * (-0,021)$) = -0,006	Neg = -0,006
			Positif			Pos = ($0,3 * (-0,021) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
			Positif			Pos = ($0,3 * (-0,021) + (0,7 * 0,9)$) = 0,624	Pos = 0,624
		T3	Negatif	-0,010	0,1	Neg = ($0,3 * (-0,010)$) = -0,003	Neg = -0,003
			Positif			Pos = ($0,3 * (-0,010) + (0,7 * 0,1)$) = 0,067	Pos = 0,067
			Positif			Pos = ($0,3 * (-0,010) + (0,7 * 0,1)$) = 0,067	Pos = 0,067
			Negatif			Neg = ($0,3 * (-0,010)$) = -0,003	Neg = -0,003
			Positif			Pos = ($0,3 * (-0,010) + (0,7 * 0,1)$) = 0,067	Pos = 0,067
			Positif			Pos = ($0,3 * (-0,010) + (0,7 * 0,1)$) = 0,067	Pos = 0,067
20%	80%	T1	Negatif	-0,020	0,9	Neg = ($0,2 * (-0,020)$) = -0,004	Neg = -0,004
			Positif			Pos = ($0,2 * (-0,020) + (0,8 * 0,9)$) = 0,716	Pos = 0,716
			Positif			Pos = ($0,2 * (-0,020) + (0,8 * 0,9)$) = 0,716	Pos = 0,716
			Negatif			Neg = ($0,2 * (-0,020)$) = -0,004	Neg = -0,004
			Positif			Pos = ($0,2 * (-0,020) + (0,8 * 0,9)$) = 0,716	Pos = 0,716
			Positif			Pos = ($0,2 * (-0,020) + (0,8 * 0,9)$) = 0,716	Pos = 0,716

α	β	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan	Hasil Penggabungan
	$(1-\alpha)$					Pos = $(\alpha * (\text{ntekstual})) + (\beta * (\text{nnon-tekstual}))$ Neg = $(\alpha * (\text{ntekstual}))$	
20%	80%	T2	Positif	-0,021	0,7	Pos = $(0,2 * (-0,020)) + (0,8 * 0,9) = 0,716$	Pos = 0,716
			Negatif			Neg = $(0,2 * (-0,021)) = -0,004$	Neg = -0,004
			Positif			Pos = $(0,2 * (-0,021)) + (0,8 * 0,9) = 0,556$	Pos = 0,556
			Positif			Pos = $(0,2 * (-0,021)) + (0,8 * 0,9) = 0,556$	Pos = 0,556
			Negatif			Neg = $(0,2 * (-0,021)) = -0,004$	Neg = -0,004
			Positif			Pos = $(0,2 * (-0,021)) + (0,8 * 0,9) = 0,556$	Pos = 0,556
			Positif			Pos = $(0,2 * (-0,021)) + (0,8 * 0,9) = 0,556$	Pos = 0,556
		T3	Negatif	-0,010	0,1	Neg = $(0,2 * (-0,010)) = -0,002$	Neg = -0,002
			Positif			Pos = $(0,2 * (-0,010)) + (0,8 * 0,1) = 0,078$	Pos = 0,078
			Positif			Pos = $(0,2 * (-0,010)) + (0,8 * 0,1) = 0,078$	Pos = 0,078
			Negatif			Neg = $(0,2 * (-0,010)) = -0,002$	Neg = -0,002
			Positif			Pos = $(0,2 * (-0,010)) + (0,8 * 0,1) = 0,078$	Pos = 0,078
			Positif			Pos = $(0,2 * (-0,010)) + (0,8 * 0,1) = 0,078$	Pos = 0,078
			Positif			Pos = $(0,2 * (-0,010)) + (0,8 * 0,1) = 0,078$	Pos = 0,078
10%	90%	T1	Negatif	-0,020	0,9	Neg = $(0,1 * (-0,020)) = -0,002$	Neg = -0,002
			Positif			Pos = $(0,1 * (-0,020)) + (0,9 * 0,9) = 0,808$	Pos = 0,808
			Positif			Pos = $(0,1 * (-0,020)) + (0,9 * 0,9) = 0,808$	Pos = 0,808
			Negatif			Neg = $(0,1 * (-0,020)) = -0,002$	Neg = -0,002
			Positif			Pos = $(0,1 * (-0,020)) + (0,9 * 0,9) = 0,808$	Pos = 0,808
			Positif			Pos = $(0,1 * (-0,020)) + (0,9 * 0,9) = 0,808$	Pos = 0,808
			Positif			Pos = $(0,1 * (-0,020)) + (0,9 * 0,9) = 0,808$	Pos = 0,808
		T2	Negatif	-0,021	0,7	Neg = $(0,1 * (-0,021)) = -0,002$	Neg = -0,002
			Positif			Pos = $(0,1 * (-0,021)) + (0,9 * 0,9) = 0,628$	Pos = 0,628
			Positif			Pos = $(0,1 * (-0,021)) + (0,9 * 0,9) = 0,628$	Pos = 0,628
			Negatif			Neg = $(0,1 * (-0,021)) = -0,002$	Neg = -0,002

α	β	Dok. Uji	Kategori	Nilai SVM	Bobot Retweet	Perhitungan Penggabungan	Hasil Penggabungan
	$(1-\alpha)$					Pos = $(\alpha * (\text{ntekstual})) + (\beta * (\text{nnon-tekstual}))$ Neg = $(\alpha * (\text{ntekstual}))$	
10%	90%	T2	Positif	-0,021	0,7	Pos = $(0,1 * (-0,021)) + (0,9 * 0,9) = 0,628$	Pos = 0,628
			Positif			Pos = $(0,1 * (-0,021)) + (0,9 * 0,9) = 0,628$	Pos = 0,628
		T3	Negatif	-0,010	0,1	Neg = $(0,1 * (-0,010)) = -0,001$	Neg = -0,001
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
			Negatif			Neg = $(0,1 * (-0,010)) = -0,001$	Neg = -0,001
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
			Positif			Pos = $(0,1 * (-0,010)) + (0,9 * 0,1) = 0,089$	Pos = 0,089
0%	100%	T1	Negatif	-0,020	0,9	Neg = $(0 * (-0,020)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,020)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Positif			Pos = $(0 * (-0,020)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Negatif			Neg = $(0 * (-0,020)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,020)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Positif			Pos = $(0 * (-0,020)) + (1 * 0,9) = 0,900$	Pos = 0,900
		T2	Negatif	-0,021	0,7	Neg = $(0 * (-0,021)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,021)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Positif			Pos = $(0 * (-0,021)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Negatif			Neg = $(0 * (-0,021)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,021)) + (1 * 0,9) = 0,900$	Pos = 0,900
			Positif			Pos = $(0 * (-0,021)) + (1 * 0,9) = 0,900$	Pos = 0,900
		T3	Negatif	-0,010	0,1	Neg = $(0 * (-0,010)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,010)) + (1 * 0,1) = 0,100$	Pos = 0,100
			Positif			Pos = $(0 * (-0,010)) + (1 * 0,1) = 0,100$	Pos = 0,100
			Negatif			Neg = $(0 * (-0,010)) = -0,000$	Neg = -0,000
			Positif			Pos = $(0 * (-0,010)) + (1 * 0,1) = 0,100$	Pos = 0,100
			Positif			Pos = $(0 * (-0,010)) + (1 * 0,1) = 0,100$	Pos = 0,100

Setelah melakukan proses penggabungan, didapatkan adanya perbedaan hasil pada klasifikasi data yang terdapat di data uji 1, data uji 2 dan data uji 3. Sebelumnya data uji 1, data uji 2 dan data uji 3 masuk ke dalam klasifikasi negatif. Namun, saat dilakukan proses penggabungan dengan menggunakan bobot

retweet, keseluruhan data uji tersebut masuk ke dalam klasifikasi positif ketika konstanta nilai $\alpha=0$ dan $\beta=1$. Hasil tersebut diperoleh dari hasil nilai maksimum atau nilai terbesar yang terdapat pada hasil penggabungan. Tabel 4.29 menunjukkan hasil yang didapatkan pada proses penggabungan pembobotan teks hasil SVM (tekstual), dengan pembobotan *retweet*.

Tabel 4.29 Klasifikasi Hasil Penggabungan

Hasil Penggabungan		Uji 1	Uji 2	Uji 3
Sebelum Penggabungan	100%, 0%	Negatif	Negatif	Negatif
Hasil setelah Penggabungan pada nilai α, β	90%, 10%	Negatif	Positif	Positif
	80%, 20%	Negatif	Positif	Positif
	70%, 30%	Negatif	Positif	Positif
	60%, 40%	Negatif	Positif	Positif
	50%, 50%	Negatif	Positif	Positif
	40%, 60%	Negatif	Positif	Positif
	30%, 70%	Negatif	Positif	Positif
	20%, 80%	Negatif	Positif	Positif
	10%, 90%	Negatif	Positif	Positif
	0%, 100%	Positif	Positif	Positif

4.8 Perancangan Pengujian

Perancangan pengujian merupakan tahap untuk melakukan pengujian terhadap sistem untuk mengetahui *persentase* nilai akurasi *opinion mining* menggunakan metode *Support Vector Machine* dan pembobotan jumlah *retweet*. Beberapa pengujian diimplementasikan yaitu sebagai berikut:

4.8.1 Perancangan pengujian pengaruh parameter pada algoritme *Support Vector Machine (SVM)* terhadap hasil akurasi

Pengujian ini untuk mengetahui pengaruh parameter pada algoritme *Support Vector Machine (SVM)* terhadap hasil akurasi. Pengujian dilakukan dengan menggunakan perbandingan rasio 70:30 yaitu 70 data latih dengan 30 data uji pada 318 data. Terdapat beberapa parameter untuk melakukan pengujian pengaruh parameter, yaitu parameter γ (*learning rate*), λ (*lambda*), C (*complexity*), dan ε (*epsilon*) menggunakan *Polynomial Kernel degree 2*.

Pada Tabel 4.30, Tabel 4.31, Tabel 4.32, dan Tabel 4.33 menunjukkan perancangan pengaruh parameter γ (*learning rate*), λ (*lambda*), C (*complexity*), dan ε (*epsilon*).

Tabel 4.30 Perancangan pengujian parameter γ (learning rate)

Nilai γ	Accuracy	Precision	Recall	F-measure
0,0001				
0,001				
0,01				
0,1				
0,0025				
0,0005				
0,005				
0,5				

Tabel 4.31 Perancangan pengujian parameter λ (lambda)

Nilai λ	Accuracy	Precision	Recall	F-measure
0,1				
0,3				
0,5				
0,7				
0,9				
1				
2				
3				

Tabel 4.32 Perancangan pengujian parameter C (complexity)

Nilai C	Accuracy	Precision	Recall	F-measure
0,0001				
0,001				
0,01				
0,1				
1				
10				
20				

Tabel 4.33 Perancangan pengujian parameter ε (epsilon)

Nilai λ	Accuracy	Precision	Recall	F-measure
0.00001				
0.0001				
0.001				
0.01				
0.1				
1				

4.8.2 Perancangan pengujian pengaruh pembobotan jumlah *retweet* terhadap hasil akurasi

Pada pengujian ini untuk mengetahui pengaruh fitur pembobotan banyaknya jumlah *retweet* dengan menggunakan perbandingan nilai konstanta α dan β sejumlah 11 kali, agar menemukan perbandingan terbaik untuk konstanta tersebut. Konstanta α akan dikalikan dengan nilai pembobotan teks (hasil klasifikasi SVM). Untuk konstanta β akan dikalikan dengan pembobotan jumlah *retweet*. Pada Tabel 4.34 menunjukkan perancangan pengujian pengaruh pembobotan jumlah *retweet*.

Tabel 4.34 Perancangan pengujian pengaruh pembobotan jumlah *retweet* terhadap hasil akurasi

No.	Konstanta		Kategori Kelas (Manual)	Kategori Kelas (Sistem)	Akurasi
	α	β (1- α)			
1	100%	0%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
2	90%	10%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
3	80%	20%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
4	70%	30%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
5	60%	40%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	

6	50%	50%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
7	40%	60%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
8	30%	70%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
9	20%	80%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
10	10%	90%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	
11	0%	100%	Negatif	Negatif	
			Negatif	Negatif	
			Positif	Positif	

4.8.3 Perancangan pengujian pengaruh *Support Vector Machine* dan pembobotan jumlah *retweet* terhadap hasil akurasi

Pengujian tahap ini akan melakukan pengujian untuk mengetahui hasil akurasi pada sistem melalui perbandingan tingkat akurasi sistem yang berdasarkan perbandingan tingkat akurasi pada pembobotan tekstual, pembobotan non-tekstual dan penggabungan.

Tabel 4.35 Perancangan pengujian pengaruh *Support Vector Machine* dan pembobotan jumlah *retweet* terhadap hasil akurasi

	Dengan <i>Support Vector Machine</i>	Dengan pembobotan jumlah <i>retweet</i>
<i>Accuracy</i>		
<i>Precision</i>		
<i>Recall</i>		
<i>F-Measure</i>		

4.9 Implementasi Sistem

Pada sub bab ini akan membahas berupa tahapan dari *opinion mining* terhadap analisis masyarakat terhadap impor beras 2018 dengan menggunakan metode *support vector machine* dan pembobotan jumlah *retweet*. Meliputi beberapa proses seperti *pre processing* teks, meliputi tokenisasi, *cleaning*, *case folding*, *filterisasi*, *stemming*. Tahap berikutnya melakukan proses pembobotan teks menggunakan TF, Wtf, DF, IDF dan TF-IDF. Selanjutnya melakukan proses klasifikasi SVM, yang terdiri dari beberapa tahapan proses yaitu matriks hessian, *sequential training*, hitung bias, hitung *sign h(x)*. Tahap yang dilakukan berikutnya yaitu menghitung penggabungan antara hasil klasifikasi SVM (tekstual) dan pembobotan jumlah *retweet* (non-tekstual). Proses yang dilakukan dengan menghitung nilai normalisasi jumlah *retweet*. Setelah memperoleh jumlah *retweet* maka dapat diketahui suatu nilai bobot, bobot tersebut akan digabungkan pada pembobotan teks untuk menentukan klasifikasi sentimen seperti kategori positif atau negatif. Tabel 4.36 menunjukkan beberapa fungsi yang akan diimplementasikan.

Tabel 4.36 Daftar fungsi implementasi sistem

NO.	Proses	Nama Fungsi	Keterangan
1.	<i>Preprocessing</i> Teks	<i>tokenisasi()</i>	Memecah suatu teks menjadi kumpulan kata-kata.
		<i>cleaning()</i>	Menghapuskan unsur-unsur pada pemrosesan teks seperti <i>username</i> , <i>link</i> url, tanda <i>retweet</i> , dan karakter <i>hashtag</i> beserta menghilangkan karakter selain huruf (<i>delimitier</i>).
		<i>case_folding()</i>	Mengubah seluruh teks menjadi huruf kecil (<i>lowercase</i>)
		<i>filterisasi()</i>	Menghapuskan kata pada dokumen yang terdapat di dalam <i>file stoplist.txt</i> .
		<i>stemming()</i>	Mengubah kata kata yang berimbuhan menjadi kata dasar.
2.	Pembobotan Teks	<i>tf()</i>	Menghitung nilai tf yang terdapat di dokumen teks.

No.	Proses	Nama Fungsi	Keterangan
2.	Pembobotan Teks	<i>wtf()</i>	Menghitung nilai wtf pada jumlah fitur kata di dokumen teks.
		<i>idf()</i>	Menghitung banyaknya dokumen yang terdapat suatu fitur kata dan menghitung nilai idf pada jumlah dokumen teks.
		<i>tf_idf()</i>	Menghitung hasil perkalian tf dan idf pada fitur kata.
3.	Klasifikasi SVM	<i>matrix_hessian()</i>	Menghitung nilai matriks hessian berdasarkan perhitungan tf-idf yang telah diproses sebelumnya.
		<i>hitung_bias()</i>	Melakukan perhitungan bias dengan menggunakan nilai α_i yang terbesar pada setiap kelas
		<i>sequential_training()</i>	melakukan perhitungan E_i , $\delta\alpha_i$, dan α_i
		<i>hitung_hx()</i>	melakukan perhitungan $Sign(h(x))$
4.	Pembobotan Jumlah Retweet	<i>normalisasi()</i>	Menghitung nilai normalisasi pada jumlah retweet yang terdapat dalam dokumen tweet.
5.	Penggabungan Pembobotan Tekstual dan Non-Tekstual	<i>hasil_penggabungan()</i>	Menghitung penggabungan pembobotan tekstual dan non-tekstual dengan menggunakan kontanta perkalian tertentu.
6.	Pengujian	<i>evaluasi()</i>	menghitung nilai <i>confusion matrix</i> , <i>precision</i> , <i>recall</i> dan <i>f-measure</i> dari hasil keluaran sistem.

4.10 Preprocessing Teks

Preprocessing teks merupakan tahap awal untuk melakukan pemrosesan teks. Tahap ini, meliputi beberapa proses yaitu tokenisasi, *cleaning*, *case folding*, filterisasi, dan *stemming*.

4.10.1 Implementasi Fungsi Tokenisasi

Fungsi *tokenisasi()* merupakan tahap untuk melakukan proses pemecahan teks pada dokumen yang berupa kumpulan kalimat menjadi kata-kata. Kode Program 4.1 menunjukkan Implementasi tokenisasi.

```
1 def tokenisasi(self,data):
2     tknzs = nltk.tokenize.TweetTokenizer()
3     hasil_token=[tknzs.tokenize(text) for text in data]
4     return hasil_token
```

Kode Program 4.1 Implementasi Tokenisasi

Penjelasan dari cuplikan Kode Program 4.1 pada implementasi tokenisasi yaitu:

1. Baris 1 merupakan definisi fungsi tokenisasi() dengan parameter (self,data). parameter data mengacu pada *tweet* data latih dan data uji.
2. Baris 2 untuk memecah dokumen teks *tweet* menjadi kata dengan pemisah karakter spasi melalui library nltk.
3. Baris 3-4 merupakan variabel hasil_token untuk menyimpan hasil tokenisasi, kemudian nilai hasil_token akan dikembalikan untuk dilakukan proses selanjutnya.

4.10.2 Implementasi Fungsi Cleaning

Fungsi *cleaning()* untuk menghapuskan unsur-unsur yang terdapat dalam dokumen teks, yang di dalam dokumen tersebut tidak mempunyai pengaruh apapun terhadap proses klasifikasi, Kode Program 4.2 menunjukkan implementasi *cleaning*.

```
1 def cleaning(self,data):
2     regex=re.compile(r'^a-zA-Z0-9_||[0-9]')
3     hasil=[]
4     for i in range(len(data)):
5         temp_baris=[]
6         for j in range(len(data[i])):
7             if "@" in data[i][j]:
8                 continue
9             if "http" in data[i][j]:
10                 continue
11             if "#" in data[i][j]:
```

Kode Program 4.2 Implementasi Cleaning

Keterangan dari cuplikan Kode Program 4.2 pada implementasi *cleaning* yaitu:

12	data[i][j]=data[i][j].replace("#","")
13	tanda_angka=regex.search(data[i][j])
14	if tanda_angka:
15	continue
16	temp_baris.append(data[i][j])
17	hasil.append(temp_baris)
18	return hasil

1. Baris 1 merupakan definisi fungsi `cleaning()` dengan parameter `(self,data)`. parameter data mengacu pada *tweet* data latih dan data uji.
2. Baris 2 membuat objek regex dan mengkompilasi pola agar mengembalikan objek regex. untuk regex `[^a-zA-Z0-9_]| [0-9]` berfungsi untuk mencocokkan karakter underscore selain huruf kecil, huruf besar, angka dan digunakan untuk menemukan karakter apa pun di antara tanda kurung. Lalu menyimpannya di dalam variabel regex.
3. Baris 3 melakukan inisialisasi variabel `hasil` sebagai array kosong untuk menyimpan nilai regex.
4. Baris 4 dan 5 merupakan merepresentasi perulangan untuk melakukan iterasi berdasarkan indeks array dalam list data kemudian di simpan pada variabel `temp_baris`.
5. Baris 7 dan 8 melakukan pendeteksian apakah terdapat tanda '@' pada list data
6. Baris 9 dan 10 melakukan pendeteksian apakah terdapat link atau 'http' pada list data.
7. Baris 11 dan 12 melakukan pendeteksian apakah terdapat tanda hashtag '#' pada list. Jika terdapat tanda hashtag '#' maka tanda tersebut dihapus dari kata tersebut.
8. Baris 13 dan 14 menunjukkan proses untuk menemukan kecocokan atau melakukan pencarian teks yang dimiliki oleh objek regex pada list data dan melakukan pendeteksian apakah terdapat tanda_angka.
9. Baris 16-18 menampilkan hasil data yang sudah di hapus pada kalimat yang mengandung '@', '#', 'http' dan tanda_angka. Lalu disimpan pada baris 16-17.

4.10.3 Implementasi Fungsi *Case Folding*

Pada fungsi `case_folding()` merupakan proses untuk mengubah keseluruhan dokumen teks menjadi huruf kecil. Kode Program 4.3 menunjukkan implementasi *case folding*.

1	def case_folding(self,data):
2	hasil =[[word.lower() for word in line] for line in
3	data]
4	return hasil

Kode Program 4.3 Implementasi *Case Folding*

Penjelasan dari cuplikan Kode Program 4.3 pada implementasi *case folding* yaitu:

1. Baris 1 merupakan definisi fungsi `case_folding()` dengan parameter (`self,data`) yang merujuk pada *tweet* data latih dan data uji.
2. Baris 2 melakukan proses untuk mengubah dokumen teks menjadi huruf kecil.

4.10.4 Implementasi Fungsi Filterisasi

Pada fungsi *filterisasi()* melakukan proses penghapusan *stopword* yang tersimpan dalam *stoplist.txt*. Kode Program 5.4 . Implementasi stoplist dan filterisasi.

```

1 def filterisasi(self, data):
2     hasil =[[word for word in line if word not in
3 self.stopword] for line in data]
4     return hasil

```

Kode Program 4.4 Implementasi *Stoplist* dan *Filterisasi*

Penjelasan dari potongan Kode Program 4.4 pada implementasi *stoplist* dan *filterisasi* yaitu:

1. Baris 1 merupakan definisi fungsi *filterisasi()* dengan parameter (`self,data`). parameter data merujuk pada *tweet* data latih dan data uji.
2. Baris 2-4 melakukan proses untuk menghapus kata tidak penting yang terdapat dalam file *stoplist.txt*.

4.10.5 Implementasi Fungsi Stemming

Pada fungsi *stemming()* untuk mengubah kata berimbuhan menjadi kata dasar dengan menggunakan library *sastrawi*. Kode Program 4.5 Implementasi *stemming*.

```

1 def stemming(self,data):
2     factory = StemmerFactory()
3     stemmer = factory.create_stemmer()
4     hasil=[]
5     i=1
6     for line in data:
7         print("baris ",i)
8         i+=1
9         temp=' '.join(line)
10        hasil_stemp= stemmer.stem(temp)
11        hasil.append(hasil_stemp.split( ))
12    return hasil

```

Kode Program 4.5 Implementasi *Stemming*

Penjelasan dari cuplikan Kode Program 4.5 pada implementasi *stemming* yaitu:

1. Baris 1 merupakan definisi fungsi *stemming()* dengan parameter (`self,data`).
2. Baris 2 melakukan proses untuk meng-*import* kelas *StemmerFactory* dari library *sastrawi*.

3. Baris 3 melakukan proses untuk membuat *stemmer* dari *library sastrawi*.
4. Baris 4 melakukan inisialisasi variabel hasil sebagai array kosong untuk menyimpan isi *file stemming*.
5. Baris 5-8 merupakan proses untuk mencetak baris pada *file txt* dan melakukan perulangan. Kemudian pada baris 9 menggabungkan nilai string pada elemen iterasi. Kemudian pada baris 10-11 merupakan proses untuk menyimpan nilai hasil stemming yang dilakukan sebelumnya ke dalam array.

4.11 Pembobotan Teks

Pembobotan teks merupakan tahapan untuk menghitung bobot kata dalam dokumen teks setelah tahap *pre processing* dilakukan.

4.11.1 Implementasi Fungsi TF

Pada fungsi *tf()* akan melakukan perhitungan nilai TF. TF merupakan banyaknya jumlah kata pada dokumen. Kode Program 5.6 menunjukkan Implementasi TF.

1	<code>def tf(self,dataset):</code>
2	<code> fitur = self.get_fitur(dataset)</code>
3	<code> return [[(dataset[i].count(x)) if dataset[i].</code>
4	<code>count(x) > 0 else 0 for x in fitur]for i in</code>
5	<code>range(len(dataset))]</code>
6	

Kode Program 4.6 Implementasi TF

Penjelasan dari cuplikan Kode Program 4.6 pada implementasi TF yaitu:

1. Baris 1 merupakan definisi fungsi *tf()* dengan parameter (*self,dataset*).
2. Baris 2 melakukan proses untuk memanggil *method get_fitur* dengan parameter *dataset*.
3. Baris 3-5 melakukan proses perhitungan *tf* untuk menghitung frekuensi kemunculan suatu kata pada hasil *pre processing*.

4.11.2 Implementasi Fungsi WTF

Pada fungsi *wtf()* menghitung bobot *log* frekuensi menggunakan nilai TF yang sudah diproses sebelumnya Kode Program 4.7 menunjukkan Implementasi WTF.

1	<code>def wtf(self,data_tf):</code>
2	<code> return [(1+math.log10(x)) if int(x) > 0 else 0 for x</code>
3	<code>in data_tf[i]]for i in range(len(data_tf))]</code>

Kode Program 4.7 Implementasi WTF

Penjelasan dari cuplikan Kode Program 4.7 pada implementasi WTF yaitu:

1. Baris 1 merupakan definisi fungsi *wtf()* dengan parameter (*self,data_tf*).

- Baris 2-3 menunjukkan kondisi untuk menghitung nilai wtf dari hasil perhitungan wtf dengan kondisi jika kondisi nilai lebih dari 0 maka melakukan perhitungan dengan algoritme yang terdapat dalam baris 2, namun apabila nilai tidak lebih dari 0 maka mendapatkan nilai 0.

4.11.3 Implementasi Fungsi IDF

Pada fungsi *idf()* merupakan frekuensi kemunculan kata pada dokumen. Implementasi IDF terdapat dalam Kode Program 4.8.

```

1      def idf(self, data_tf):
2          temp_idf=[0 for x in range(len(data_tf))]
3          for i in range(len(data_tf)):
4              for j in range(len(data_tf[i])):
5                  if data_tf[i][j]>0:
6                      temp_idf[i]+=1
7          return [math.log10(len(data_tf[0])/x) for x in temp_
8      idf]
```

Kode Program 4.8 Implementasi IDF

Penjelasan dari cuplikan Kode Program 4.8 pada implementasi IDF yaitu:

- Baris 1 merupakan definisi fungsi *idf()* dengan parameter (self, data_tf).
- Baris 2-8 melakukan perhitungan nilai DF untuk menghitung frekuensi munculnya kata pada dokumen. Kemudian baris 7-8 proses algoritme untuk menghitung nilai IDF dari nilai DF.

4.11.4 Implementasi Fungsi TF-IDF

Pada fungsi *tfidf()* melakukan perhitungan perkalian pada nilai TF dan IDF yang didapatkan sebelumnya. Kode Program 4.9 Implementasi TF-IDF.

```

1      def tf_idf(self, data_wtf, data_idf):
2          tf_idf= [(data_wtf[i][x] * data_idf[x]) for x in
3          range(len(data_wtf[0]))]for i in range (len(data_wtf))]
4          return tf_idf
```

Kode Program 4.9 Implementasi TF-IDF

Penjelasan dari potongan Kode Program 4.9 pada implementasi TF-IDF yaitu:

- Baris 1 merupakan definisi fungsi *tf_idf()* dengan parameter (self, data_wtf, data_idf).
- Baris 2-3 merupakan proses untuk menghitung nilai TF-IDF untuk menghitung nilai Wtf pada suatu kata, kemudian nilai IDF dikalikan dengan nilai Wtf. Lalu nilai TF-IDF disimpan pada baris 4.

4.12 Klasifikasi SVM

Klasifikasi teks SVM merupakan bagian tahapan yang diproses setelah tahapan selesai dilakukan sebelumnya. Pada tahap ini, meliputi beberapa proses perhitungan yaitu Matriks Hessian, *Sequential Training*, Hitung Bias, dan Hitung $Sign(h(x))$.

4.12.1 Implementasi Matriks Hessian

Pada fungsi *matrix_hessian()* akan dilakukan proses perhitungan nilai matriks Hessian. *Kernel* yang digunakan yaitu *kernel polynomial* berderajat dua. Implementasi matriks Hessian terdapat dalam Kode Program 4.10.

1 2 3 4 5 6 7	<pre>def matrix_hessian(self, lamda, kelas, matrix_kernel): matrix_hessian=[[None for _ in range(len(matrix_ kernel))] for _ in range(len(matrix_kernel))] for x in range(len(matrix_kernel)): for y in range(len(matrix_kernel)): matrix_hessian[x][y]=kelas[x]*kelas[y]* (matrix_kernel[x][y]+pow(lamda,2)) return matrix_hessian;</pre>
---------------------------------	--

Kode Program 4.10 Implementasi Matriks Hessian

Penjelasan dari cuplikan Kode Program 4.10 pada implementasi Matriks Hessian yaitu:

1. Baris 1 merupakan definisi fungsi *matrix_hessian()* dengan parameter (self, lamda, kelas, matrix_kernel).
2. Baris 2-3 melakukan proses perulangan pada variabel *matrix_kernel*
3. Baris 4-6 melakukan proses perhitungan *matrix_hessian* dengan menggunakan *kernel polynomial degree 2*.

4.12.2 Implementasi *Sequential Training*

Pada fungsi *sequential_training* akan dilakukan proses perhitungan nilai $\delta\alpha_i$ dan α_i serta nilai E_i hingga mencapai iterasi batas maksimum yaitu $\max(|\delta\alpha_i|) < \varepsilon$. Apabila sudah mencapai batas maksimum maka iterasi akan berhenti. Kode Program 4.11. menunjukkan Implementasi hitung *sequential_training*.

```

1  def hitung_Ei(self, alpha, matrix_hessian):
2      Ei=[0 for x in range(len(alpha))]
3      for i in range(len(matrix_hessian)):
4          for j in range(len(matrix_hessian)):
5              Ei[i]+=matrix_hessian[j][i]*alpha[i]
6      return Ei
7
8  def delta_alpa(self, learning_rate, arr_alpha, C, Ei):
9      return [min(max(learning_rate*(1-Ei[j]), -arr_
10 alpha[j]), C-arr_alpha[j]) for j in range(len(Ei))]
11
12 def update_alpha(self, alpha, delta):
13     return [alpha[i]+delta[i] for i in range(len(alpha))]
14
15 def sequential_training(self, C, itermax, epsilon, alpha,
16 learning_rate, matrix_hessian):
17     kondisi=False
18     arr_alpha=[alpha for x in range(len(matrix_hessian))]
19     for i in range(itermax):
20         print("iterasi"+str(i))
21         Ei=self.hitung_Ei(arr_alpha, matrix_hessian)
22         delta_alpa=self.delta_alpa(learning_rate, arr_
23 alpha, C, Ei)
24         arr_alpha=self.update_alpha(arr_alpha, delta_
25 alpha)
26         max_delta= max(delta_alpa)
27         print(max_delta)
28         if max_delta<epsilon:
29             break
30     return arr_alpha

```

Kode Program 4.11 Implementasi Sequential Training

Penjelasan dari potongan Kode Program 4.11 pada implementasi *Sequential Training* yaitu:

1. Baris 1 merupakan definisi fungsi `hitung_Ei()` dengan parameter (`self`, `lamda`, `kelas`, `matrix_hessian`).
2. Baris 2 melakukan proses perulangan dan menyimpan variabel `alpha`.
3. Baris 3-6 melakukan proses perulangan dan menyimpan variabel `matriks_hessian`. Kemudian melakukan algoritme perhitungan nilai `Ei` yang terdapat baris 5 yaitu nilai `alpha` dikalikan dengan hasil nilai `matriks_hessian` sebelumnya.
4. Baris 8 merupakan definisi fungsi `delta_alpa` dengan parameter (`self`, `learning_rate`, `arr_alpha`, `C`, `Ei`).
5. Baris 9-10 melakukan proses menghitung $\delta\alpha_i$ yang terdapat pada Persamaan 2.8.
6. Baris 12 merupakan definisi fungsi `delta_alpa` dengan parameter (`self`, `alpha`, `delta`).
7. Baris 13 melakukan proses menghitung α_i yang terdapat pada Persamaan 2.9.

8. Baris 15-16 merupakan definisi fungsi `sequensial_training` dengan parameter (`self`, `C`, `itermax`, `epsilon`, `alpha`, `learning_rate`, `matrix_hessian`).
9. Baris 16-20 untuk membuat objek `arr_alpha` untuk melakukan proses perulangan nilai `alpha` dan menyimpan nilai variabel *matriks hessian* serta memproses perulangan sesuai batas nilai maksimum.
10. Baris 21 untuk membuat objek E_i untuk memanggil fungsi `hitung_Ei` dengan parameter (`arr_alpha`, `matrix_hessian`).
11. Baris 22-23 untuk membuat objek `delta_alpha` untuk memanggil fungsi `delta_alpha` dengan parameter (`learning_rate`, `arr_alpha`, `C`, E_i).
12. Baris 24-25 untuk membuat objek `arr_alpha` untuk memanggil fungsi `update_alpha` dengan parameter (`arr_alpha`, `delta_alpha`).
13. Baris 26-30 untuk membuat objek `max_delta` yang mempunyai parameter `delta_alpha`. Kemudian melakukan kondisi jika nilai `delta` sudah mencapai nilai batas maksimum maka iterasi akan berhenti.

4.12.3 Implementasi Hitung Bias

Pada fungsi `hitung_bias()` akan dilakukan proses perhitungan nilai bias yang terdapat pada Persamaan 2.10. Kode Program 4.12 menunjukkan Implementasi hitung bias.

```

1  def hitung_bias(self, arr_alpha, kelas, tf_idf,
2  learning_rate, d):
3      bobot_neg=0
4      bobot_pos=0
5      index_max_neg=self.candidatMin(arr_alpha, kelas)
6      index_max_pos=self.candidatMax(arr_alpha, kelas)
7      kernel_neg=self.hitung_kernel_pos_neg(index_max_
8  neg, tf_idf, d, c)
9      kernel_pos=self.hitung_kernel_pos_neg(index_max_
10 pos, tf_idf, d, c)
11     kernel_neg=kernel[index_max_neg]
12     kernel_pos=kernel[index_max_pos]
13     for j in range(len(arr_alpha)):
14         bobot_neg+=arr_alpha[j]*learning_rate*
15     kernel_neg[j]
16         bobot_pos+=arr_alpha[j]*learning_rate*
17     kernel_pos[j]
18     return -(bobot_pos+bobot_neg)/2
19 
```

Kode Program 4.12 Implementasi Hitung Bias

Penjelasan dari cuplikan Kode Program 4.12 pada implementasi Hitung Bias yaitu:

1. Baris 1-2 merupakan definisi fungsi `hitung_bias()` dengan parameter (`self`, `arr_alpha`, `kelas`, `tf_idf`, `learning_rate`, `d`).
2. Baris 3 melakukan proses inisialisasi `bobot_neg` bernilai 0.
3. Baris 4 melakukan proses inisialisasi `bobot_pos` bernilai 0.

4. Baris 5 membuat objek `index_max_neg` untuk melakukan pemanggilan fungsi `candidatMin` dengan parameter (`arr_alpha`, `kelas`).
5. Baris 6 membuat objek `index_max_pos` untuk melakukan pemanggilan fungsi `candidatMax` dengan parameter (`arr_alpha`, `kelas`).
6. Baris 7-8 membuat objek `kernel_neg` untuk melakukan pemanggilan fungsi `hitung_kernel_pos_neg` dengan parameter (`index_max_neg`, `tf_idf`, `d`, `c`)
7. Baris 9-10 membuat objek `kernel_pos` untuk melakukan pemanggilan fungsi `hitung_kernel_pos_neg` dengan parameter (`index_max_pos`, `tf_idf`, `d`, `c`)
8. Baris 11 membuat objek `kernel_neg` untuk melakukan pemanggilan objek `index_max_neg` yang disimpan dalam bentuk array.
9. Baris 12 membuat objek `kernel_pos` untuk melakukan pemanggilan objek `index_max_positif` yang disimpan dalam bentuk array .
10. Baris 13-18 melakukan proses perulangan untuk memproses perhitungan nilai bobot negatif pada nilai α yang terbesar di kelas negatif yang terdapat di baris 14-15 dan memproses perhitungan nilai bobot positif pada nilai α yang terbesar di kelas positif yang terdapat di baris 16-17. Kemudian menghitung nilai bias dengan menggunakan algoritme yang terdapat di baris 14.

4.12.4 Implementasi Hitung $Sign(h(x))$

Pada fungsi `hitung_hx()` atau hitung kemiripan yaitu melakukan proses perhitungan nilai untuk memasukkan data uji ke *hyperplane* untuk mengetahui nilai kelas sentimen pada Persamaan 2.11. Implementasi hitung kemiripan terdapat dalam Kode Program 5.13.

<pre> 1 2 3 4 5 6 7 8 9 10 </pre>	<pre> def hitung_hx(self,bias, alpha, kelas, data_uji,tf_idf,c): temp=0 kernel=self.hitung_kernel_uji(data_uji,tf_idf,2,c) for i in range(len(alpha)): temp+=(alpha[i]*kelas[i]*kernel[i]) hx=temp+bias if hx>=0: return 1 else: return -1 </pre>
-----------------------------------	--

Kode Program 4.13 Implementasi Hitung Kemiripan

Penjelasan dari cuplikan Kode Program 4.13 pada implementasi Hitung Kemiripan yaitu:

1. Baris 1 merupakan definisi fungsi `hitung_hx()` dengan parameter (`self`, `bias`, `alpha`, `kelas`, `data_uji`, `tf_idf`, `c`).
2. Baris 2-3 membuat objek `kernel` untuk memanggil fungsi `hitung_kernel_uji` untuk diproses pada baris berikutnya dengan parameter (`data_uji`, `tf_idf`, `2`, `c`).

- Baris 4-10 melakukan proses perulangan dan melakukan proses perhitungan dengan menggunakan algoritme yang terdapat pada baris 5. Apabila proses perhitungan selesai dilakukan maka selanjutnya memproses perhitungan yang terdapat pada Persamaan 2.11. Pada baris 7 menggambarkan kondisi jika nilai hx atau nilai kemiripan lebih dari 0 maka menghasilkan nilai sebaliknya jika nilai hx atau nilai kemiripan kurang dari 0 maka menghasilkan nilai -1.

4.13 Implementasi Fungsi Normalisasi Jumlah *Retweet*

Pembobotan jumlah *retweet* suatu penambahan fitur untuk melakukan proses klasifikasi teks. Tahap ini akan menghitung nilai normalisasi Min-max pada banyaknya suatu *retweet* yang didapatkan pada dokumen. Bobot didapatkan banyaknya suatu *retweet*. Selanjutnya lakukan proses hitung normalisasi Min-max. Pada fungsi *normalisasi()*, melakukan hitung normalisasi jumlah *retweet* yang terdapat pada masing masing *tweet* untuk menghasilkan bobot pada banyaknya *retweet* tersebut. Kode Program 4.14 menunjukkan Implementasi pembobotan jumlah *retweet*.

1 2 3 4 5 6 7 8	<pre>def normalisasi_rt(self, data, x): print(x) print(data) datas=data datas.append(x) mins=min(datas) maxs=max(datas) return (x-mins)/(maxs-mins)*(0.9-0.1)+0.1</pre>
--------------------------------------	---

Kode Program 4.14 Implementasi Normalisasi Jumlah *Retweet*

Penjelasan dari potongan Kode Program 4.14 pada implementasi Normalisasi jumlah *retweet* yaitu:

- Baris 1 merupakan definisi fungsi *normalisasi_rt()* dengan parameter (self, data, x).
- Baris 2 menampilkan jumlah *retweet* yang terdapat dalam data latih dan data uji.
- Baris 4-7 melakukan proses untuk mengetahui hasil nilai maksimal dan nilai minimum pada jumlah *retweet*.
- Baris 8 melakukan proses perhitungan nilai normalisasi Min-max.

4.14 Implementasi Penggabungan Pembobotan Tekstual dan Non-Tekstual

Implementasi ini melakukan proses hitung untuk mengetahui apakah sistem berjalan dengan baik. Dengan mengimplementasikan pengujian *confusion matrix* dengan menggunakan parameter *accuracy*, *precision*, *recall*, dan *f-measure* agar mengetahui seberapa baiknya sistem yang dibangun.

```

1 print("proses penggabungan")
2 a=0#,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0#
3 b=1#,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1
4 tempGabungan=[]
5 gabungan=[]
6 for j in range(len(kelas_uji)):
7     for k in range(len(kelas_latih)):
8         if kelas_latih[j]==-1:
9             tempGabungan.append(a*hasil_svm[j])
10            elif kelas_latih[j]==1:
11                tempGabungan.append((a*hasil_svm[j])+(b*bobot_retweet
12 _uji[j]))
13            gabungan.append(max(tempGabungan))
14

```

Kode Program 4.15 Implementasi Penggabungan Pembobotan Teksual dan Non-Teksual

Penggabungan dari potongan Kode Program 4.15 pada implementasi Pembobotan Teksual dan Non-Teksual yaitu:

1. Baris 1-3 merupakan proses inisialisasi nilai konstanta alpha dan beta untuk melakukan proses perhitungan penggabungan.
2. Baris 4-5 untuk menyimpan nilai dalam array.
3. Baris 6-13 melakukan proses perulangan untuk menghitung nilai penggabungan yang melibatkan nilai suatu konstanta pada alpha dan beta. Baris 8-9 jika kelas pada data latih bernilai negatif maka nilai alpha dikalikan dengan hasil klasifikasi teks SVM yang didapatkan sebelumnya. Baris 10-12 dilakukan jika kelas pada data latih bernilai positif maka nilai konstanta alpha akan dikalikan dengan hasil svm yang ditambah dengan hasil perkalian jumlah *retweet* dan konstanta beta.
4. Baris 13 mengetahui nilai kelas suatu dokumen pada hasil yang diperoleh melalui perhitungan penggabungan. Nilai paling maksimal pada perhitungan penggabungan tersebut akan menjadi penentuan nilai kelas.

4.15 Pengujian

Pengujian menggunakan parameter *accuracy*, *precision*, *recall*, dan *f-measure* yang termasuk pada bagian *confusion matrix*. Untuk mengetahui seberapa baik sistem yang dibangun. Proses pengujian ini dilakukan agar mengetahui tingkat keberhasilan dari sistem.

4.15.1 Implementasi Fungsi Evaluasi

Pada fungsi *evaluasi* melakukan proses hitung *confusion matrix* untuk mengetahui nilai *true positive*, *true negatif*, *false positive*, dan *false negatif*. Parameter *akurasi* untuk proses hitung nilai akurasi terhadap hasil keluaran sistem. Parameter *precision* untuk melakukan proses perhitungan nilai *precision* terhadap hasil keluaran sistem. Parameter *recall* proses perhitungan nilai *recall* terhadap hasil keluaran sistem dan Parameter *fmeasure* akan dilakukan proses

perhitungan nilai *fmeasure* terhadap hasil keluaran sistem. Kode Program 4.16 Implementasi fungsi *evaluasi*.

```

1  def evaluasi(self, y, hasil):
2      conf_matrix=[[0 for x in range(2)] for y in range(2)]
3      for i in range(len(y)):
4          if y[i]==1 and hasil[i]==1:
5              conf_matrix[0][0]+=1 #tp
6          elif y[i]==-1 and hasil[i]==1:
7              conf_matrix[1][0]+=1 #tn
8          elif y[i]==1 and hasil[i]==-1:
9              conf_matrix[0][1]+=1 #fp
10         elif y[i]==-1 and hasil[i]==-1:
11             conf_matrix[1][1]+=1 #fn
12         akurasi=(conf_matrix[0][0]+conf_matrix[1][1])/
13         (conf_matrix[0][0]+conf_matrix[0][1]+conf_matrix[1][0]
14         +conf_matrix[1][1])
15         precision=(conf_matrix[0][0])/(conf_matrix[0][0]+
16         conf_matrix[1][0])
17         recall=(conf_matrix[0][0])/(conf_matrix[0][0]+
18         conf_matrix[0][1])
19         f_measure=2*precision*recall/(precision+recall)
20         return akurasi, precision, recall, f_measure, conf_matrix

```

Kode Program 4.16 Implementasi Evaluasi

Penjelasan dari potongan Kode Program 4.16 pada implementasi *evaluasi* yaitu:

1. Baris 1-20 melakukan proses pengecekan pada nilai klasifikasi sistem dengan nilai klasifikasi sebenarnya dalam data uji. Apabila ketika proses pengecekan benar dan document *tweet* termasuk sentimen positif maka lakukan perhitungan dan penjumlahan pada variabel *tp* dan jika dokumen *tweet* termasuk sentiment negatif maka lakukan perhitungan dan penjumlahan variabel *tn*. Pada saat lakukan pengecekan terdapat data uji yang salah yang seharusnya termasuk dokumen *tweet* sentimen positif maka akan dijumlahkan pada variabel *fn*, selanjutnya apabila dokumen tweet termasuk sentimen negatif maka akan dijumlahkan pada variabel *fp*.

BAB 5 HASIL DAN PEMBAHASAN

Pada bagian yang menjabarkan hasil pengujian dan pembahasan akan menggunakan metode *Support Vector Machine* terhadap parameter parameter yang sudah ditentukan. Serta hasil nilai akurasi terhadap penambahan fitur pembobotan jumlah *retweet*. Data latih yang digunakan pada pengujian ini sejumlah 222 data *tweet* yang terdiri dari 111 *tweet* kelas positif dan 111 *tweet* kelas negatif serta data uji sejumlah 96 data *tweet* yang terdiri dari 48 *tweet* kelas positif dan 48 *tweet* kelas negatif.

5.1 Skenario Pengujian

Sub bab ini berfungsi untuk mengetahui nilai presentase akurasi yang dihasilkan oleh sistem. Dengan melakukan pengujian untuk mengetahui pengaruh nilai parameter γ (*learning rate*), λ (*lambda*), C (*complexity*) dan ϵ (*epsilon*) pada algoritme *Support Vector Machine*. Pengujian selanjutnya yang dilakukan yaitu pengujian untuk mengetahui pengaruh pembobotan jumlah *retweet*. Kemudian pada tahap akhir pengujian ini akan dilakukan pengujian untuk mengetahui nilai akurasi pada algoritme *Support Vector Machine* dan pembobotan jumlah *retweet*.

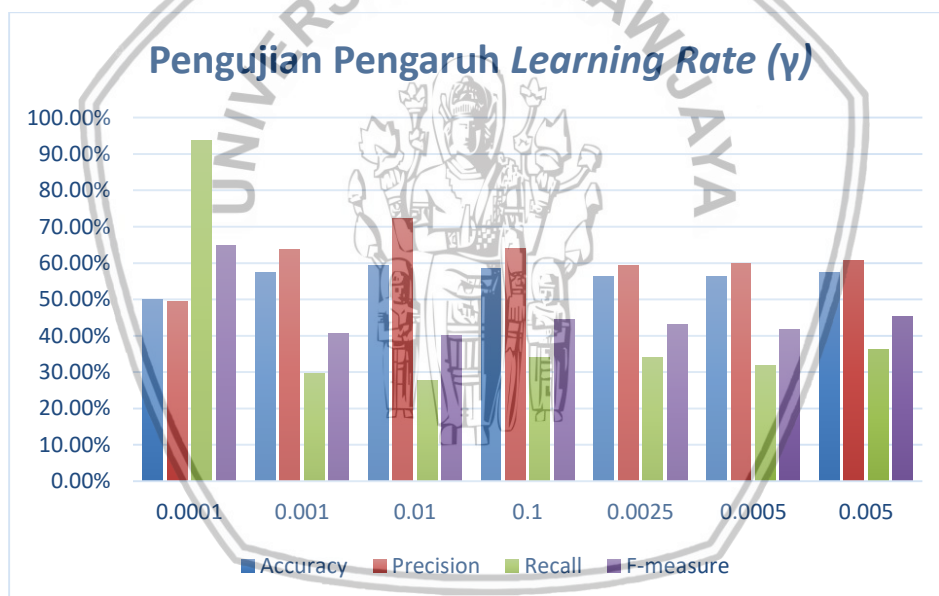
5.1.1 Pengujian Nilai Parameter γ (*learning rate*)

Pengujian ini akan dilakukan pengujian untuk mengetahui pengaruh nilai parameter γ (*learning rate*) dengan menggunakan *kernel polynomial* berderajat dua, λ (*lambda*) = 0,5, C (*complexity*) = 0,01, ϵ (*epsilon*) = 0,00001 dan banyaknya iterasi = 50. Tabel 5.1 menampilkan hasil pengaruh nilai parameter γ (*learning rate*) pada akurasi yang diperoleh. Dengan mengimplementasikan perbandingan rasio data latih dan data uji sebesar 70:30. Hasil dari pengujian dapat dilihat pada Tabel 5.1.

Tabel 5.1 Hasil pengujian nilai parameter γ (*learning rate*)

Nilai γ	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
0,0001	50,00%	49,43%	93,61%	64,70%
0,001	57,29%	63,63%	29,78%	40,57%
0,01	59,37%	72,22%	27,65%	40,00%
0,1	58,33%	64,00%	34,04%	44,45%
0,0025	56,25%	59,25%	34,04%	43,24%
0,0005	56,25%	60,00%	31,91%	41,67%
0,005	57,29%	60,71%	36,17%	45,33%
0,5	57,29%	61,53%	34,04%	43,83%

Berdasarkan pengujian yang telah dilakukan bahwa akurasi tertinggi didapatkan pada nilai parameter *learning rate* sebesar 0,01 dengan memperoleh nilai *accuracy* sebesar 59,37%. Pada nilai *learning rate* = 0,1 hasil akurasi mulai mengalami konvergen. Nilai *learning rate* digunakan untuk membentuk himpunan *support vector* dan melakukan proses hitung nilai $\delta\alpha$, nilai ini berfungsi untuk lakukan hitung pada berhentinya kondisi iterasi. Apabila nilai tersebut semakin kecil maka menyebabkan nilai $\text{Max}(|\delta\alpha|) < \varepsilon$ maka kondisi iterasi berhenti jika nilai $\delta\alpha$ dibawah epsilon karena nilai sudah konvergen. Pada nilai *precision* memperoleh hasil sebesar 72,22% menunjukkan kesesuaian informasi yang diinginkan dengan yang diberikan oleh sistem sudah memiliki nilai yang cukup tinggi. Sedangkan pada nilai *recall* memperoleh nilai yang cukup tinggi yaitu 93,61 pada nilai *learning rate* = 0,0001, maka menandakan bahwa sistem sudah baik dalam menemukan kembali informasi. Selanjutnya, nilai *f-measure* tertinggi yang didapatkan pada pengujian ini sebesar 64,70%. Nilai tinggi dan rendahnya *f-measure* ditentukan dari nilai *precision* serta nilai *recall* yang didapatkan. Gambar 5.1 menunjukkan hasil pengujian pengaruh nilai parameter γ (*learning rate*). Nilai γ (*learning rate*) = 0,01 akan digunakan untuk proses selanjutnya.



Gambar 5.1 Pengaruh nilai parameter γ (*learning rate*)

5.1.2 Pengujian Nilai Parameter λ (*lambda*)

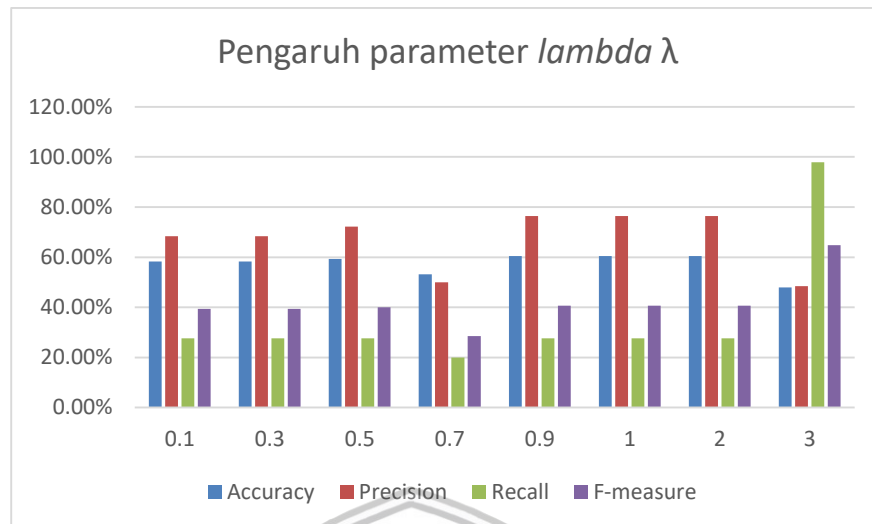
Pada skenario ini akan dilakukan pengujian untuk mengetahui pengaruh nilai parameter λ (*lambda*). Dengan menggunakan nilai *learning rate* yang didapatkan pada pengujian sebelumnya yaitu 0,01, $C = 0,01$, ε (*epsilon*) = 0,00001, dengan iterasi = 50. Tabel 5.2 menunjukkan hasil akurasi pada nilai parameter λ (*lambda*) yang didapatkan.

Tabel 5.2 Hasil pengujian nilai parameter parameter λ (*lambda*)

Nilai λ	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
0,1	58,33%	68,42%	27,65%	39,39%
0,3	58,33%	68,42%	27,65%	39,39%
0,5	59,37%	72,22%	27,65%	40,00%
0,7	53,12%	50,00%	20,00%	28,57%
0,9	60,41%	76,47%	27,65%	40,62%
1	60,41%	76,47%	27,65%	40,62%
2	60,41%	76,47%	27,65%	40,62%
3	47,91%	48,42%	97,87%	64,78%

Hasil pengujian yang telah dilakukan bahwa akurasi tertinggi didapatkan pada nilai parameter *lambda* (λ) sebesar 60,41%. Nilai *lambda* (λ) yang semakin besar akan berpengaruh pada tingkat hasil akurasi karena berpengaruh ketika melakukan proses perhitungan Matriks Hessien. Perhitungan tersebut akan mengalami kelambatan saat kecepatan mencapai kovergensi apabila nilai *lambda* (λ) semakin besar dan berpengaruh pada ketidakstabilannya proses *learning* (Vijayakumar & Wu, 1999). Pada nilai *precision* memperoleh hasil sebesar 76,47%. Maka hasil tersebut sudah cukup baik dalam mengukur ketepatan sistem. Semakin tinggi nilai *accuracy* maka akan memengaruhi perolehan nilai *precision*. Sedangkan pada nilai *recall* memperoleh nilai yang cukup tinggi yaitu 97,87 pada nilai *learning rate* = 3. Hasil tersebut menunjukkan bahwa tingkat keberhasilan sistem sudah baik dalam mengklasifikasi data yang diprediksi dan yang diperoleh pada kelas positif. Saat *learning rate* berada di nilai 0,1 – 2 hasil *recall* yang didapatkan tidak tinggi sebab ditentukan oleh besarnya hasil nilai *accuracy* dan *precision*. Semakin besarnya nilai *accuracy* dan *precision* maka nilai *recall* akan menjadi rendah, sebaliknya apabila nilai *accuracy* dan *precision* rendah maka nilai *recall* akan menjadi tinggi. Selanjutnya, nilai *f-measure* tertinggi yang didapatkan pada pengujian ini sebesar 64,78%. Nilai tinggi dan rendahnya *f-measure* ditentukan dari nilai *precision* serta nilai *recall* yang didapatkan. Nilai *f-measure* merupakan rata-rata nilai yang serupa antara nilai *precision* dan *recall*.

Pada Gambar 5.2 menunjukkan hasil yang didapatkan pada parameter λ (*lambda*). Nilai λ (*lambda*) = 0,9 merupakan hasil terbaik yang diperoleh pada pengujian ini. Nilai λ (*lambda*) tersebut akan diimplementasikan pada pengujian berikutnya karena nilai tersebut memiliki akurasi terbaik yang diujikan.



Gambar 5.2 Pengaruh nilai parameter λ

5.1.3 Pengujian Nilai Parameter C (*complexity*)

Pengaruh pengujian nilai untuk parameter ini untuk mengetahui pengaruh nilai C (*complexity*). Dengan mengimplementasikan parameter γ (*learning rate*) dan λ (*lambda*) yang didapatkan dari perolehan nilai sebelumnya. Pengujian pengaruh nilai parameter C menggunakan nilai ϵ (*epsilon*) = 0,00001 dan nilai iterasi = 50. Tabel 5.3 menunjukkan hasil pengaruh nilai parameter C.

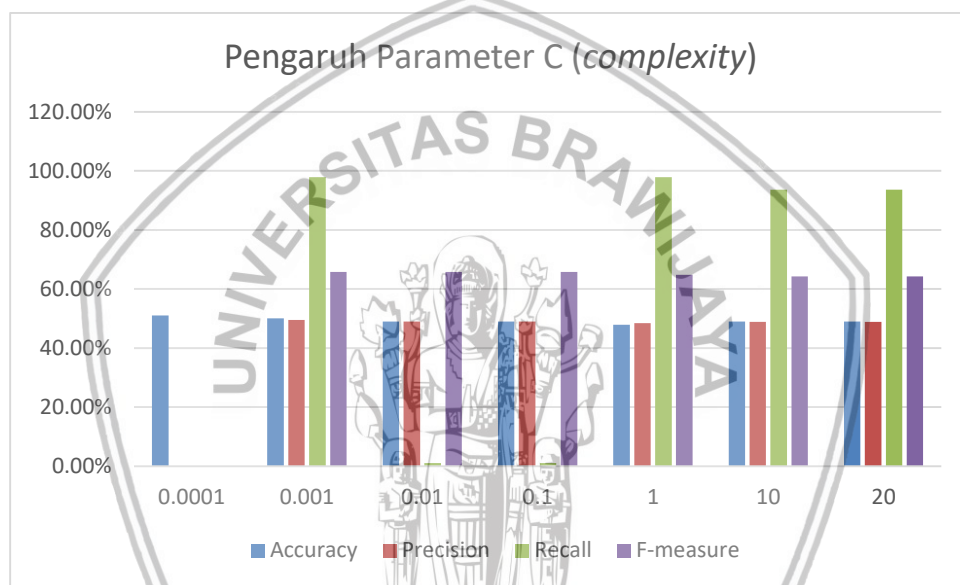
Tabel 5.3 Hasil Pengujian Nilai Parameter C (*complexity*)

Nilai C	Accuracy	Precision	Recall	F-measure
0,0001	51,04%	Tak terhitung	Tak terhitung	Tak terhitung
0,001	50,00%	49,46%	97,87%	65,71%
0,01	48,95%	48,95%	01,00%	65,73%
0,1	48,95%	48,95%	01,00%	65,73%
1	47,91%	48,42%	97,87%	64,78%
10	48,95%	48,88%	93,61%	64,23%
20	48,95%	48,88%	93,61%	64,23%

Tabel 5.4 menunjukkan hasil pengujian yang telah dilakukan didapatkan akurasi tertinggi memperoleh nilai λ sebesar 51,04%. Nilai akurasi semakin rendah apabila saat kondisi nilai C (*complexity*) semakin tinggi. Nilai C (*complexity*) berpengaruh ketika melakukan proses perhitungan $\delta\alpha$. $\delta\alpha$ merupakan nilai variabel tunggal untuk mencari nilai α terbaru ditambah dengan nilai α sebelumnya dan terhadap pencarian *support vector*. Sebab *support vector* memberikan pengaruh pada nilai bias b, karena nilai bias membantu melakukan proses klasifikasi dokumen. Pada nilai *precision* memperoleh hasil sebesar 49,46%. Namun terdapat nilai tak terhitung atau 0 ketika nilai C = 0,0001 karena hasil *true*

positive (TP) dan *false positive* (FP) yang didapatkan memperoleh nilai 0, maka jika *true positive* (TP) dan *false positive* (FP) dibagi dengan 0 akan menyebabkan nilai tak terhingga atau 0, begitupun yang terdapat pada nilai *recall* dan *f-measure* pada pengujian tersebut. Nilai *recall* memperoleh nilai yang cukup rendah yaitu 1 saat *learning rate* berada di nilai 0,01 dan 0,1 dikarenakan pengaruh nilai nilai *true positive* (TP) dan *false negatif* yang didapatkan pada hasil *confusion matrix*. Selanjutnya, nilai *f-measure* tertinggi yang didapatkan pada pengujian ini sebesar 65,71%. Nilai tinggi dan rendahnya *f-measure* ditentukan dari nilai *precision* serta nilai *recall* yang didapatkan.

Gambar 5.3 pengujian parameter C (*complexity*) diperoleh hasil akurasi terbaik saat C (*complexity*) bernilai 0,001 dengan akurasi sebesar 50,00. Nilai C (*complexity*) tersebut akan diimplementasikan pada pengujian tahap berikutnya.



Gambar 5.3 Pengaruh nilai parameter C (*complexity*)

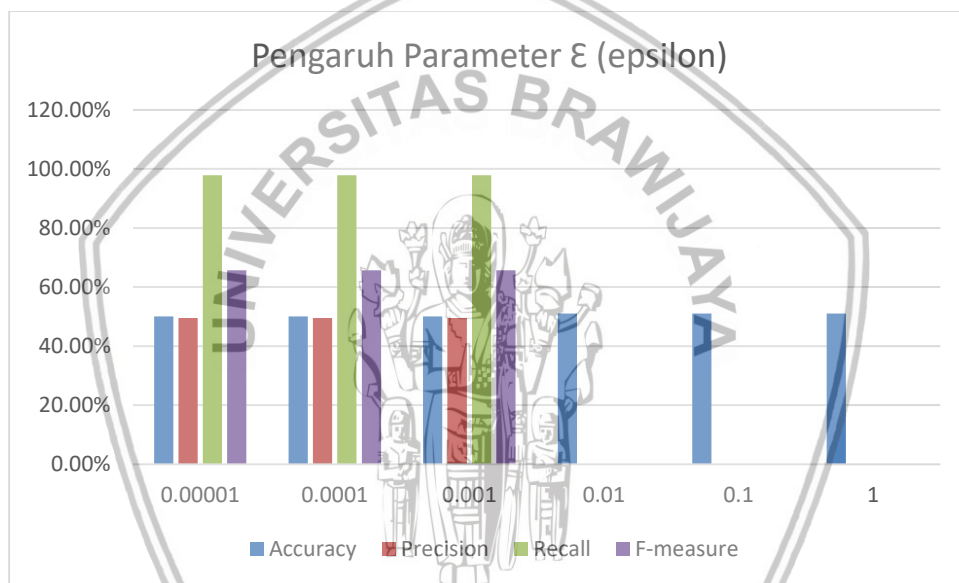
5.1.4 Pengujian Nilai Parameter ϵ (*epsilon*)

Bagian ini untuk mengetahui pengaruh nilai ϵ (*epsilon*). Pengujian dilakukan dengan menggunakan nilai γ (*learning rate*), λ (*lambda*) dan C (*complexity*) yang diimplementasikan pada pengujian sebelumnya. Pengujian nilai parameter ϵ (*epsilon*) menggunakan nilai iterasi = 50. Tabel 5.4 menunjukkan hasil pengujian parameter ϵ (*epsilon*).

Tabel 5.4 Hasil Pengujian Nilai Parameter ϵ (*epsilon*)

ϵ (<i>epsilon</i>)	Accuracy	Precision	Recall	F-measure
0.00001	50,00%	49,46%	97,87%	65,71%
0.0001	50,00%	49,46%	97,87%	65,71%
0.001	50,00%	49,46%	97,87%	65,71%
0.01	51,04%	Tak terhingga	Tak terhingga	Tak terhingga
0.1	51,04%	Tak terhingga	Tak terhingga	Tak terhingga
1	51,04%	Tak terhingga	Tak terhingga	Tak terhingga

Hasil pengujian pengaruh parameter ϵ (*epsilon*) memperoleh hasil terbaik pada ϵ (*epsilon*) = 0,001 dengan akurasi sebesar 51.04%. Nilai ϵ (*epsilon*) untuk mengetahui perubahan yang terjadi pada nilai α . Apabila $\max(|\delta\alpha|) < \epsilon$, akan terjadi suatu perubahan yang tidak terlalu signifikan. Terdapat nilai tak terhingga atau 0 pada nilai *accuracy*, *precision*, *recall* dan *f-measure* ketika nilai *epsilon* 0,01 – 1 dikarenakan hasil *true positive* (TP) dan *false positive* (FP) yang didapatkan memperoleh nilai 0, maka jika *true positive* (TP) dan *false positive* (FP) dibagi dengan 0 akan menyebabkan nilai tak terhingga atau 0. Gambar 5.4 menunjukkan hasil pengujian pengaruh parameter ϵ (*epsilon*) yang memperoleh hasil akurasi terbaik pada parameter ϵ (*epsilon*) = 0,0001 dengan akurasi sebesar 50,00. Nilai seluruh parameter terbaik yang sudah dilakukan pengujian sebelumnya akan diimplementasikan pada tahap pengujian berikutnya.



Gambar 5.4 Pengaruh nilai parameter ϵ (*epsilon*)

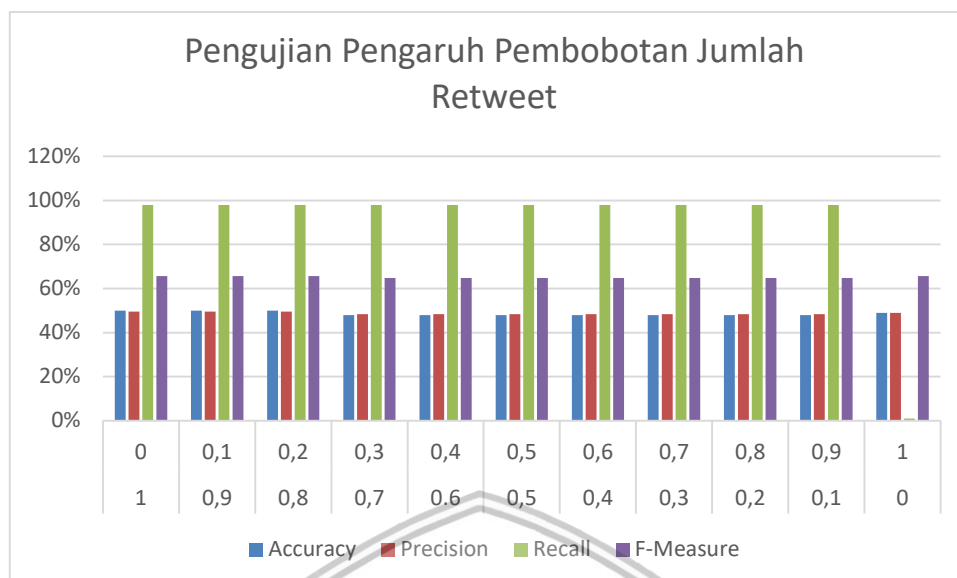
5.1.5 Pengujian Pengaruh Pembobotan Jumlah *Retweet*

Bagian ini untuk mengetahui pengaruh pada penambahan fitur pembobotan jumlah *retweet*. Pengujian ini akan mengimplementasikan nilai parameter terbaik dalam metode SVM pada pengujian yang dilakukan sebelumnya. Bagian ini akan melakukan proses hitung normalisasi Min-max terlebih dahulu untuk mendapatkan nilai bobot *retweet* berdasarkan jumlah *retweet* yang diperoleh dari data latih dan data uji. Serta menggunakan nilai parameter yang terbaik pada pengujian sebelumnya. Nilai konstanta α dan β digunakan sebagai perbandingan nilai untuk mengetahui hasil terbaik sejumlah 11 kali untuk melakukan proses pembobotan jumlah *retweet*. Konstanta α yaitu konstanta untuk melakukan perhitungan perkalian pada hasil nilai bobot teks SVM. Konstanta β yaitu konstanta untuk melakukan perhitungan perkalian pada bobot jumlah *retweet*. Tabel 5.5 menunjukkan hasil pengaruh pembobotan jumlah *retweet*.

Tabel 5.5 Hasil Pengujian Pembobotan Jumlah Retweet

No.	Konstanta		<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
	α (alpha)	β (beta)				
1	1	0	50,00%	49,46%	97,87%	65,71%
2	0,9	0,1	50,00%	49,46%	97,87%	65,71%
3	0,8	0,2	50,00%	49,46%	97,87%	65,71%
4	0,7	0,3	47,91%	48,42%	97,87%	64,78%
5	0,6	0,4	47,91%	48,42%	97,87%	64,78%
6	0,5	0,5	47,91%	48,42%	97,87%	64,78%
7	0,4	0,6	47,91%	48,42%	97,87%	64,78%
8	0,3	0,7	47,91%	48,42%	97,87%	64,78%
9	0,2	0,8	47,91%	48,42%	97,87%	64,78%
10	0,1	0,9	47,91%	48,42%	97,87%	64,78%
11	0	1	48,95%	48,95%	01,00%	65,73%

Hasil pengujian yang ditunjukkan pada Tabel 5.5 bahwa hasil evaluasi terbaik diperoleh ketika nilai konstanta $\alpha = 1$, $\beta = 0$ hingga ketika nilai konstanta $\alpha = 0,8$, $\beta = 0,2$ dengan nilai *accuracy* sebesar 50,00%, nilai *precision* 49,46%, nilai *recall* 97,87% dan nilai *f-measure* 65,71%. Pada perolehan hasil evaluasi ketika nilai $\alpha = 0,7$, $\beta = 0,3$ hingga ketika nilai konstanta $\alpha = 0,1$, $\beta = 0,9$ dengan nilai *accuracy* sebesar 47,91%, nilai *precision* 48,42%, nilai *recall* 97,87% dan nilai *f-measure* 64,78%. Saat nilai konstanta $\alpha = 0$, $\beta = 1$ mendapatkan hasil nilai *accuracy* sebesar 47,91%, nilai *precision* 48,42%, nilai *recall* 01,00% dan nilai *f-measure* 65,73%. Dapat disimpulkan bahwa adanya penambahan fitur pembobotan jumlah *retweet* dapat memberikan pengaruh terhadap nilai evaluasi meskipun tidak terlalu signifikan. Selanjutnya perbandingan nilai yang berada di konstanta $\alpha = 0$, $\beta = 1$ didapatkan adanya peningkatan nilai *f-measure* seperti hasil penelitian sebelumnya, bahwa fitur ini mampu meningkatkan nilai *f-measure* (Perdana & Pinandito, 2017). Adanya penggunaan kata tidak baku dapat menjadi salah satu pengaruh dalam pengujian ini dikarenakan adanya fitur tidak relevan seperti saat melakukan proses filterisasi maupun *stemming* sehingga nilai *accuracy* yang diperoleh tidak terlalu optimal. Untuk perolehan nilai *recall* sebesar 01,00% disebabkan karena adanya kesalahan dalam memprediksi kelas atau menemukan kembali informasi yang didapatkan. Gambar 5.4 menunjukkan hasil pengujian pengaruh pembobotan jumlah *retweet* dengan pembanding nilai konstanta sebanyak 11 kali.



Gambar 5.5 Pengujian Pengaruh Pembobotan Jumlah Retweet pada perbandingan nilai konstanta sebanyak 11 kali

5.1.6 Pengujian pengaruh *Support Vector Machine* dan pembobotan jumlah *retweet* terhadap hasil akurasi

Pengujian ini akan melakukan pengujian untuk membandingkan hasil akurasi yang didapatkan ketika menggunakan metode *Support Vector Machine* tanpa fitur pembobotan jumlah *retweet* dan menggunakan metode *Support Vector Machine* dengan fitur pembobotan jumlah *retweet* dengan menggunakan nilai-nilai parameter terbaik yang di dapatkan sebelumnya. Yaitu nilai γ (*learning rate*) = 0,01, λ (*lambda*) = 0,9, C (*complexity*) = 0,001 dan ϵ (*epsilon*) = 0,0001 dengan menggunakan nilai iterasi = 50. Tabel 5.6 menunjukkan hasil *confusion matrix* yang didapatkan pada sistem dan Tabel 5.7

Tabel 5.6 *Confusion Matrix* terhadap hasil akurasi

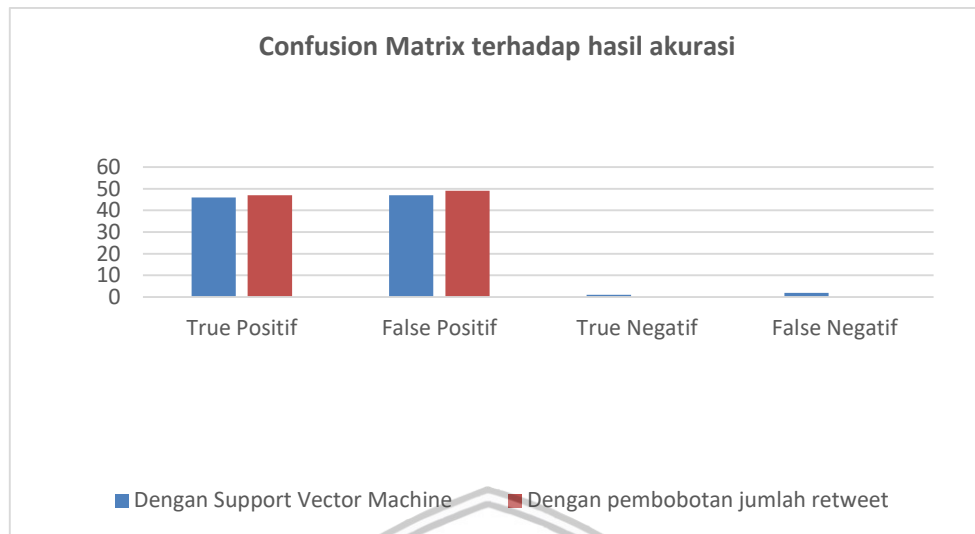
	Dengan <i>Support Vector Machine</i>	Dengan pembobotan jumlah <i>retweet</i>
True Positif	46	47
False Positif	47	49
True Negatif	1	0
False Negatif	2	0

Tabel 5.7 Pengaruh hasil akurasi dengan menggunakan pembobotan jumlah *retweet* dan tanpa menggunakan pembobotan jumlah *retweet*

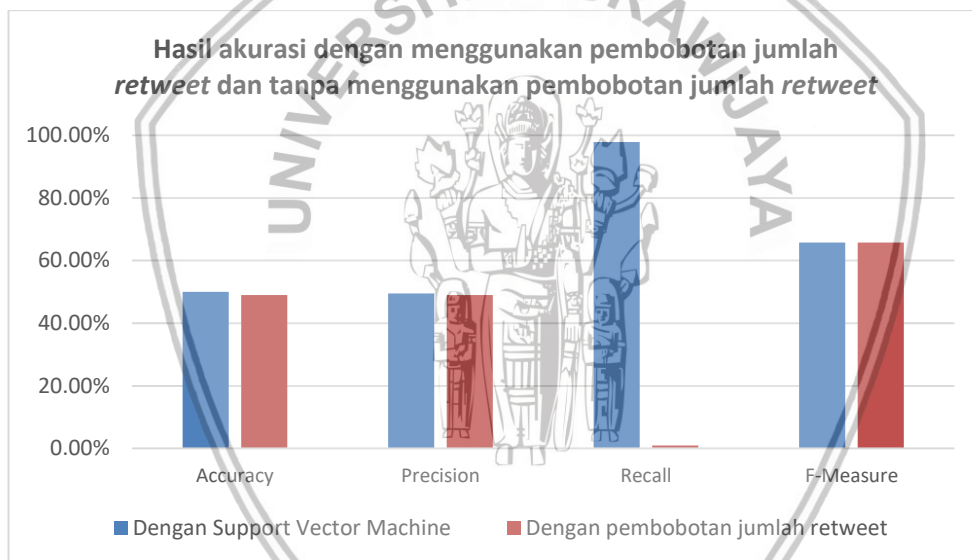
	Dengan <i>Support Vector Machine</i>	Dengan pembobotan jumlah <i>retweet</i>
Accuracy	50,00%	48,95%
Precision	49,46%	48,95%
Recall	97,87%	01,00%
F-Measure	65,71%	65,73%

Hasil yang ditunjukkan pada Tabel 5.7 bahwa nilai akurasi yang terdapat di indikator *accuracy*, *precision*, *recall*, *matrix* mendapatkan perolehan nilai *accuracy* sebesar 50,00%, nilai *precision* 49,46%, nilai *recall* 97,87% dan nilai *f-measure* 65,71% dengan menggunakan metode *Support Vector Machine* tanpa menggunakan pembobotan jumlah *retweet*. Sedangkan nilai akurasi yang terdapat di indikator *accuracy*, *precision*, *recall*, *matrix* mendapatkan perolehan nilai *accuracy* sebesar 48,95%, nilai *precision* 48,95%, nilai *recall* 01,00% dan nilai *f-measure* 65,73% dengan menggunakan metode *Support Vector Machine* dan menggunakan pembobotan jumlah *retweet*.

Berdasarkan hasil tersebut maka dapat dijabarkan bahwa dengan menggunakan SVM didapatkan false negatif 2 yang artinya kelas tersebut sebenarnya masuk pada kelas positif tetapi diprediksi masuk pada kelas negatif. Maka dengan adanya penambahan fitur akan membantu memperoleh kelas positif dan mengurangi jumlah nilai *false negatif*. Selanjutnya, hasil akurasi terendah yang didapatkan pada pengujian ini sebesar 48,95 dikarenakan adanya jumlah *retweet* pada data uji bernilai 0. Oleh sebab itu, akan diberi bobot atau nilai terendah 0,1 yang diperoleh pada perhitungan normalisasi. Serta tingginya perolehan nilai pada *false positif* sejumlah 49 sangat mempengaruhi juga di hasil *recall* yang begitu sangat rendah. Akibatnya nilai yang sebenarnya masuk pada kelas negatif tetapi diprediksi masuk pada kelas positif. Di samping itu, nilai evaluasi yang kurang optimal disebabkan pada persebaran kata yang tidak bisa dihilangkan maupun dibakukan dalam proses *pre processing* pada tahap *filtering* dan *stemming* yang terdapat di data latih dan data uji. Tetapi dengan adanya penambahan fitur mampu menambah nilai *f-measure* meskipun tidak terlalu signifikan. Gambar 5.6 menunjukkan *confusion matrix* terhadap hasil akurasi dan Gambar 5.7 menunjukkan hasil akurasi dengan menggunakan fitur dan tanpa menggunakan fitur.



Gambar 5.6 Confusion Matrix terhadap hasil akurasi



Gambar 5.7 Hasil akurasi dengan menggunakan pembobotan jumlah *retweet* dan tanpa menggunakan pembobotan jumlah *retweet*

BAB 6 KESIMPULAN DAN SARAN

Bagian ini menjabarkan mengenai kesimpulan yang didapatkan pada penelitian ini serta saran yang berguna sebagai masukan untuk penelitian sebelumnya.

6.1 Kesimpulan

Melalui hasil pengujian yang diuraikan sebelumnya. Maka didapatkan beberapa kesimpulan dalam opinion mining untuk mengetahui analisis masyarakat terhadap impor beras 2018 pada twitter dengan menerapkan metode *Support Vector Machine* dan penambahan fitur pembobotan jumlah retweet. Sebagai berikut:

1. Penerapan metode *Support Vector Machine* dapat diterapkan dalam melakukan *opinion mining* terhadap analisis sentimen impor beras 2018 pada *tweet* Twitter. Terdiri dari klasifikasi kelas positif dan kelas negatif berdasarkan analisis kelas sentimen positif dan analisis kelas sentimen negatif. Dengan menggunakan Data latih sejumlah 222 data *tweet* yang terdiri dari 111 *tweet* kelas positif dan 111 *tweet* kelas negatif serta data uji sejumlah 96 data *tweet* yang terdiri dari 48 *tweet* kelas positif dan 48 *tweet* kelas negatif pada pengujian ini. Tahap proses yang dilakukan pada penelitian ini dengan melakukan proses *pre processing* teks, perhitungan pembobotan teks dan perhitungan klasifikasi teks dengan metode *Support Vector Machine*. Selanjutnya perhitungan normalisasi *min-max* memberikan bobot pada jumlah *retweet* yang berfungsi melakukan perhitungan penggabungan melalui nilai pembobotan non-tekstual (jumlah *retweet*) dan pembobotan non-tekstual (hasil nilai klasifikasi metode SVM) untuk melakukan perkalian pada konstanta yang diinisialisasikan α dan β . Hasil yang diperoleh berupa klasifikasi kelas sentimen positif atau kelas sentimen negatif melalui nilai paling maksimum yang didapatkan pada hasil nilai penggabungan.
2. Adanya penambahan fitur pembobotan jumlah *retweet* untuk analisis sentimen ini. Mampu meningkatkan nilai *f-measure* meskipun tidak terlalu signifikan dibanding tidak menggunakan penambahan fitur. Namun di samping itu, adanya nilai evaluasi yang kurang optimal disebabkan pada persebaran kata yang tidak bisa dihilangkan maupun dibakukan dalam proses *pre processing* pada tahap *filtering* dan *stemming* yang terdapat di data latih dan data uji.
3. Analisis sentimen impor beras 2018 pada *tweet* Twitter mendapatkan perolehan nilai *accuracy* sebesar 48,95%, nilai *precision* 48,95%, nilai *recall* 01,00% dan nilai *f-measure* 65,73% menggunakan metode *Support Vector Machine* dan menggunakan pembobotan jumlah

retweet. Hal tersebut mampu melakukan klasifikasi analisis *tweet* berdasarkan analisis sentimen.

6.2 Saran

Terdapat beberapa saran terhadap penelitian selanjutnya, sebagai berikut:

1. Dapat menggunakan kata baku agar mampu meningkatkan performa klasifikasi.
2. Dapat mengimplementasikan kernel yang lain seperti additive kernel SVM.
3. Dapat menambahkan seleksi fitur yang lain selain TF-IDF dan Pembobotan Jumlah *Retweet*.
4. Dapat menggunakan metode klasifikasi lain dalam melakukan pemrosesan teks.



DAFTAR REFERENSI

- Aggarwal, C. C. & ChengXiang, Z., 2012. *Mining Text Data*. New York: Springer.
- Ahuja, R. et al., 2017. *Twitter Based Model For Emotional State Classification*. 4th IEEE International Conference on Signal Processing, Computing and Control (ISPCC 2017). [e-journal] Solan: IEEE, hal. 494-498. Tersedia di: <<https://ieeexplore.ieee.org/document/8269729>>.
- Andreas, D., 2018. *Beras Jadi Pemicu Inflasi di Januari 2018*. [online] Tersedia di: <<https://tirto.id/beras-jadi-pemicu-inflasi-di-januari-2018-cD85>> [Diakses 16 Februari 2018].
- Campbell, C. & Yiming, Y., 2011. *Learning with Support Vector Machines*. Morgan & Claypool Publishers. [e-book] Tersedia di: <<https://www.morganclaypool.com/doi/abs/10.2200/S00324ED1V01Y201102AIM010>> [Diakses 18 Juli 2018].
- Desai, M. dan Mehta, M., 2016. *Techniques For Sentiment Analysis Of Twitter Data: A Comprehensive Survey*. 2016 International Conference on Computing, Communication and Automation (ICCCA). [e-journal] Noida: IEEE, hal. 149-154. Tersedia di: <<https://ieeexplore.ieee.org/document/7813707>>.
- Estu, W., Perdana, R.S. dan Fauzi, M.A., 2017. Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode *K-Nearest Neighbor* dan Pembobotan Jumlah *Retweet*. Pengembangan Teknologi Informasi dan Ilmu Komputer, [e-journal] 1(12), hal.1750–1757. Tersedia di: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/631>>.
- Handayani, S. L., 2018. *Impor Turunkan Harga Gabah Petani*. [online] Tersedia di: <<https://www.republika.co.id/berita/ekonomi/pertanian/18amjad/09/26/pfngpd370-impor-turunkan-harga-gabah-petani>> [Diakses 2 Oktober 2018].
- Huda, R. M., 2017. *Indonesia Pengguna Twitter Terbesar Ketiga Dunia*. [online] Tersedia di: <<http://setara.net/indonesia-pengguna-twitter-terbesar-ketiga-dunia/>> [Diakses 16 April 2018].
- Junaedi, H., Budianto, H., Maryati, I. dan Melani, Y., 2011. *Data Transformation Pada Data Mining. Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi"*. [e-journal] Surabaya: IdeaTech2011, hal.93-99. Tersedia di: <http://ideatech.stts.edu/proceeding2011/12000113_INF%20Hartarto%20p93-99.pdf>
- Amjad, K., Ishtiaq, M., Firdous, S., Mehmood, A.M., 2017. Exploring Twitter News Biases Using Urdu-based Sentiment Lexicon. *International Conference on*

- Open Source Systems & Technologies (ICOSST)* [e-journal] Lahore: ICOSST, hal. 48-53. Tersedia di: < <https://ieeexplore.ieee.org/document/8279004/>>
- Kristiyanti, A. D., 2015. Analisis Sentimen Review Produk Kosmetik Menggunakan Algoritma *Support Vector Machine* dan *Particle Swarm Optimization* Sebagai Metode Seleksi Fitur. *Seminar Nasional Inovasi dan Tren (SNIT) 2015*, [e-journal] hal. A134-A141. Tersedia di: < <https://pdfs.semanticscholar.org/3390/0a9210138391b4b99095d463fd5708a422fb.pdf>>.
- Librian, A., 2014. *Sastrawi Library PHP untuk Stemming Bahasa Indonesia*. [online] Tersedia di: <<https://github.com/sastrawi/sastrawi>> [Diakses 20 Maret 2018].
- Listiyarini, T., 2016. *Naik ke Peringkat Dua Dunia, Impor Gandum RI Capai 8,1 Juta Ton*. [online] Tersedia di: <<http://www.beritasatu.com/ekonomi/337466-naik-ke-peringkat-dua-dunia-impor-gandum-ri-capai-81-juta-ton.html>> [Diakses 17 Februari 2018].
- Liu, Bing., 2012. *Sentiment Analysis and Opinion Mining*. New York: Morgan & Claypool Publishers.
- Liu, Bing., 2015. *Sentiment Analysis Mining Opinions, Sentiments, and Emotions*. New York: Cambridge University Press.
- Luqyana, A. W., Cholissodin, Imam dan Perdana, R. S., 2018. Analisis Sentimen *Cyberbullying* pada Komentar Instagram dengan Metode Klasifikasi *Support Vector Machine*. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, [e-journal] 2(11), hal.4704-4712. Tersedia di: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/3051/1195/>>
- Nugroho, S. A., Witarto, B. A. dan Handoko, D., 2003. *Support Vector Machine: Teori dan Aplikasinya dalam Bioinformatika*. [online] Tersedia di: <<http://asnugroho.net/papers/ikcsvm.pdf>> [Diakses 20 Mei 2018].
- Nursiyono, J. A., 2014. *Tetap Impor Beras Padahal Produksi Surplus, Ini Alasannya*. [Online] Tersedia di: <https://www.kompasiana.com/jokoade/tetap-impor-beras-padaahal-produksi-surplusinalasannya_54f6727ca333116a7d88b4cf8> [Diakses 16 Februari 2018].
- Pang, Bo dan Lee, Lillian, 2008. *Opinion Mining and Sentiment Analysis*. New York: Now Publishers. [e-book] Tersedia di: <<http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf>> [Diakses 1 Juli 2018].
- Perdana, R. S. & Pinandito, A., 2017. *Combining Likes-Retweet Analysis and Naive Bayes Classifier Within Twitter for Sentiment Analysis*. *International Conference On Communication And Computer Engineering*. hal.41-46. Tersedia di: <<http://journal.utem.edu.my/index.php/jtec/article/view/3732>>

- Perez, S., 2017. *Tweeting Made Easier*. [online] Tersedia di: <https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html> [Diakses 23 Maret 2018].
- Reily, M., 2018. *Bulog : 281 Ribu Ton Beras Impor Siap Datang Bertahap*. [online] Tersedia di: <<https://katadata.co.id/berita/2018/02/06/bulog-281-ribu-ton-beras-impor-siap-datang-bertahap>> [Diakses 13 Februari 2018].
- Rofiqoh, U., Perdana, R.S. dan Fauzi, M.A., 2017. Analisis Sentimen Tingkat Kepuasan Pengguna Penyedia Layanan Telekomunikasi Seluler Indonesia Pada Twitter Dengan Metode *Support Vector Machine* dan *Lexion Based Feature*. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, [e-journal] 1(12), hal.1725–1732. Tersedia di: <<http://j-ptiik.ub.ac.id/index.php/jptiik/article/view/628>>.
- Ruslan, K., 2018. *Mengapa Beras Kerap Bikin Gaduh?*. [online] Tersedia di: <<https://www.kompasiana.com/kadirsaja/5a7d46d3cf01b4165e50c6f2/mengapa-beras-kerap-bikin-gaduh>> [Diakses 16 Februari 2018].
- Srivastava, A. N. & Sahami, M., 2009. *Text Mining Classification, Clustering, and Applications*. United States of America: CRC Press.
- tala, 2003. *Stopwords collection of Bahasa Indonesia collected from many sources*. [online] Tersedia di: <<https://github.com/masdevi/ID-Stopwords>> [Accessed 5 Maret 2018].
- Vijayakumar, S. & Wu, S., 1999. Sequential Support Vector Classifiers and Regression. *International on Soft Computing*, [e-journal] Genoa:(SOCO'99), Hal.610-619 Tersedia di: <https://www.researchgate.net/publication/2318644_Sequential_Support_Vector_Classifiers_and_Regression>
- Zarella, D., 2009. *The Social Media Marketing Book*. Canada: O'Reilly Media.